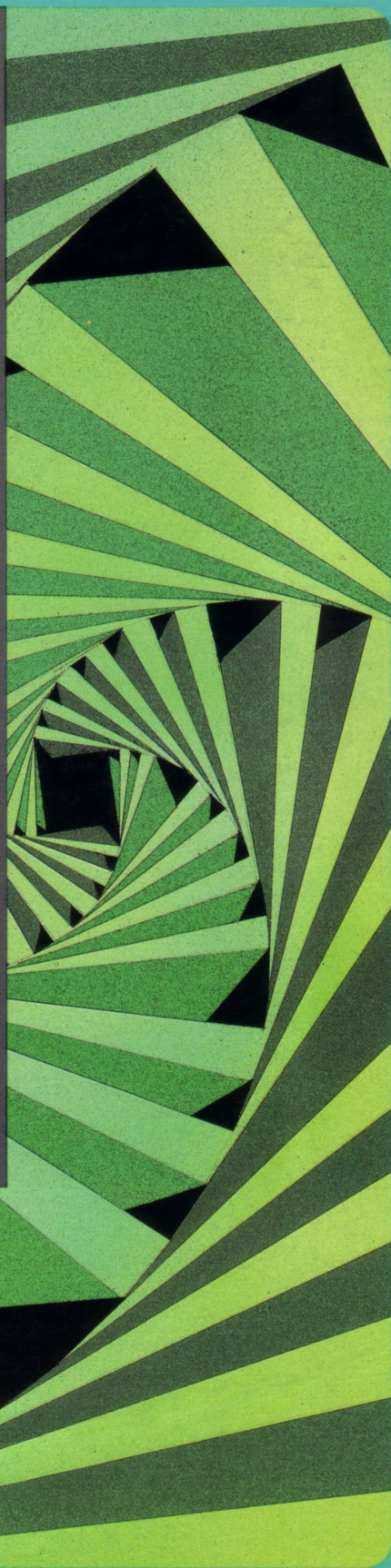
OS/2

GUIDA ALL'OPERATING SYSTEM 2

Tiziano Martelli

il piacere del computer

Il piacere del computer

1 *Tom Rugg e Phil Feldman* 32 programmi con il PET
2 *Rich Didday* Intervista sul personal computer, hardware
3 *Tom Rugg e Phil Feldman* 32 programmi con l'Apple
4 *Ken Knecht* Microsoft Basic
5 *Paul M. Chirlian* Pascal
6 *Tom Rugg e Phil Feldman* 32 programmi con il TRS-80
7 *Rich Didday* Intervista sul personal computer, software
8 *Herbert D. Peckham* Imparate il Basic con il PET/CBM
9 *Karl Townsend e Merl Miller* Il personal computer come professione
10 *Karen Billings e David Moursund* Te ne intendi di computer?
11 *T. Dwyer e M. Critchfield* Il Basic e il personal computer, uno: introduzione
12 *Don Inman e Kurt Inman* Imparate il linguaggio dell'Apple
13 *T. Dwyer e M. Critchfield* Il Basic e il personal computer, due: applicazioni
14 *Luigi Pierro* Il manuale del CP/M
15 *Carlo Sintini* A scuola con il PET/CBM
16 *David Johnson-Davies* Il manuale dell'Atom
17 *David E. Schultz* Il libro del Commodore VIC 20
18 *Jim Huffman e Robert Bruce* Il "debug" nei personal computer
19 *John M. Nevison* Programmazione in Basic per l'uomo d'affari
20 *Mark Harrison* Imparate il Basic con lo ZX81
21 *Ronald W. Anderson* Dal Basic al Pascal
22 *Herbert D. Peckham* Imparate il Basic con il Texas TI 99/44
23 *Sergio Borsani* A scuola con il Texas TI 99/4A
24 *Jerry Willis e Deborrah Willis* Come usare il Commodore 64
25 *Mark Harrison* Imparate il Basic con lo Spectrum
26 *Carlo Sintini e Costantino Mustacchio* A scuola con il Commodore 64
27 *David A. Lien* Imparate il Basic con l'IBM PC
28 *Ken Tracton* Introduzione al Lisp
29 *Fabio Mavaracchio* Programmi in Basic per l'elettronica
30 *Ian Stewart e Robin Jones* Il linguaggio macchina dello Spectrum
31 *Tom Rugg, Phil Feldman e C.S. Wilson* 32 programmi per il VIC 20
32 *Merl Miller e Mary A. Myers* Introduzione all'Apple Macintosh
33 *Stam Krute* Grafica e suoni con il Commodore 64
34 *Jerry Willis e William Manning* Come usare l'IBM PCjr
35 *Tom Rugg e Phil Feldmann* 32 programmi con il Commodore 64
36 *Sergio Borsani* Matematica e geometria con il Commodore 64
37 *David Laine* ZX Spectrum: tecniche avanzate di linguaggio macchina
38 *Salvatore Marseglia* Chimica con il pocket computer
39 *Patrizio Quintilli* Basic per i geometri
40 *Roy Atherton* Programmare in SuperBasic con il QL
41 *Carl Townsend* Il sistema operativo MSDOS
42 *Carlo Sintini e Costantino Mustacchio* Grafici di funzioni
43 *Salvatore Marseglia* Chimica con il personal computer
44 *Paul Chirlian* Programmare in Forth
45 MSX Basic, guida di riferimento *a cura della Comtrad*
46 *Lawrie Moore* Musica, grafica e programmazione per Spectrum e Spectrum Plus
47 *Giancarlo Baccolini* Progetti speciali con il Vic 20 e il C64
48 *Stanley Trost* Economia e finanza personale in Apple Basic
49 *Claudio Parmelli* Guida alle banche dati
50 *Paul M. Chirlian* Introduzione al C
51 *Douglas S. Stivison* Introduzione al Turbo Pascal
52 *Carl Townsend* Introduzione al Turbo Prolog
53 *Jonathan Kamin* La gestione dei dischi rigidi

GUIDA ALL'OPERATING SYSTEM 2

Tiziano Martelli

franco muzzio editore

Indice

Ringraziamenti

Un ringraziamento a tutte le persone che mi hanno aiutato a portare a termine quest'impresa. Un ringraziamento particolare va a Maria Pia Schiavina e a Maria Di Nicco per il contributo che mi hanno dato nella redazione e successiva correzione del testo; ancora un ringraziamento a Paolo Bondioli che è il promotore dell'iniziativa e infine a Virginio Sala che ha permesso che questo progetto si realizzasse. Un ringraziamento è dovuto alla concessionaria SIED di Bologna per il supporto informativo fornitomi nella fase di test del sistema.

Premessa

Questo volume è stato realizzato utilizzando per prove, riferimenti e maschere video riprodotte all'interno, IBM Operating System/2 versione 1.00 in lingua inglese e il Software Development Kit della Microsoft che comprende la versione 1.03 di Microsoft Operating System/2, anch'essa in inglese. La versione in lingua italiana del prodotto sarà disponibile entro Giugno 1988.

Introduzione

Sono diverse le motivazioni che hanno spinto chi scrive a decidere di preparare questa guida. Prima fra tutte è senz'altro la speranza che questo nuovo sistema operativo, che è stato addirittura progettato per offrire all'utilizzatore una condizione di lavoro il più possibile simile a quella cui è abituato, si diffonda rapidamente fra i vari produttori di software e diffonda soprattutto questa caratteristica di familiarità fra utilizzatore e prodotto.
Il limite principale dei sistemi operativi che hanno contribuito a formare l'esperienza di chi scrive, non è mai stato nella difficoltà di realizzare particolari funzioni di elaborazione ma, semmai, nelle difficoltà incontrate per spiegare le funzionalità del calcolatore e chi doveva utilizzarne le proprietà elaborative.
In effetti, sistemi operativi come UNIX, XENIX e, in misura minore, MS-DOS, sono sistemi operativi specializzati nelle funzioni di gestione delle risorse dei calcolatori: offrono una grande capacità di elaborazione ed una grande elasticità nel supporto di diversi tipi di unità periferiche come stampanti, plotter, perforatori, digitizer e tavolette grafiche, mouse, penne ottiche e così via.
Malgrado ciò, il supporto per funzioni di interfaccia ad alto livello con chi utilizza il calcolatore è sempre stato relegato ad un ruolo di secondo piano: chi lo riteneva proprio indispensabile, poteva dotare i propri programmi di questa interfaccia, tuttavia lo faceva a proprio carico e senza alcuno standard di riferimento.
Ci sono voluti anni perché questo problema ottenesse la dovuta attenzione: nel frattempo alcuni costruttori, come ad esempio Apple, hanno aperto la strada alla definizione di uno standard di interfaccia utente con prodotti come il Macintosh.
OPERATING SYSTEM/2, prodotto dalla statunitense Microsoft in seguito ad un

progetto comune con IBM, come del resto è avvenuto anche per MS-DOS, presenta un disegno completamente nuovo in merito alla realizzazione del dialogo con l'utilizzatore: anziché l'ormai decrepita linea di comando che imponeva un rapporto domanda-risposta su un argomento per volta, viene proposta una serie di maschere video, ciascuna associata ad un diverso programma o a diversi aspetti elaborativi dello stesso programma, poste in una catasta e pertanto "sfogliabili" proprio come gli appunti, le circolari e tutto ciò che noi mettiamo sulla nostra scrivania in ordine sparso ma che possiamo consultare liberamente semplicemente ponendo l'oggetto della nostra attenzione "in cima al mucchio".
Questa prima versione di Operating System/2, che porta la sigla 1.0, non possiede ancora questa interfaccia, il cui nome è WINDOW PRESENTATION MANAGER, ma offre già tutto il supporto necessario per la sua realizzazione, che sarà comunque portata a termine entro la fine del 1988, e, soprattutto, offre il supporto necessario a chi produce software affinché sia in grado fin da ora di progettare ed realizzare le proprie applicazioni future secondo questo nuovo schema.
Oltre al WINDOW PRESENTATION MANAGER, OS/2 presenta due ulteriori caratteristiche non meno importanti e significative, in modo particolare per chi proviene dall'ambiente MS-DOS: la possibilità dell'esecuzione contemporanea di più procedure e il superamento del limite della memoria RAM utilizzabile, che passa dal Megabyte di MS-DOS ai 16 Mbyte indirizzati direttamente da OS/2 fino al Gigabyte di memoria che OS/2 può utilizzare in modo virtuale.
Sempre per permettere ai produttori di software una migrazione dal DOS il più indolore possibile, OS/2 offre la possibilità di avere, per uno dei processi gestiti, la compatibilità con il DOS. In effetti, è stato anche definito un sottoinsieme delle chiamate di sistema che, se rispettato, permette di realizzare un programma utilizzabile sia in emulazione DOS che in ambiente OS/2 vero e proprio.
Questa guida si propone di essere di aiuto sia a coloro che, conoscendo il DOS, decidono di passare a questo nuovo sistema operativo sia a coloro che si accingono ad affrontare per la prima volta un sistema operativo: l'unico sforzo richiesto a questi ultimi è di saltare i riferimenti al DOS che spesso intervengono nel testo.
La struttura del volume, a partire dalla fine di questa introduzione, è esposta di seguito:

Capitolo 1 Il sistema operativo: cos'è e cosa fa.
Questo capitolo è rivolto a coloro che non conoscono alcun sistema operativo e le interazioni che intercorrono tra questo e l'hardware del computer. Chiunque utilizzi già un calcolatore conosce probabilmente gli argomenti trattati in questo capitolo e può tranquillamente saltarlo.

Capitolo 2 I sistemi operativi per personal computer: DOS e OS/2.
Questo capitolo espone le differenze fra i due sistemi operativi e i requisiti minimi che un calcolatore deve possedere per l'utilizzo di OS/2. È utile per coloro a cui occorre una veloce panoramica delle caratteristiche strutturali di OS/2.

Capitolo 3 L'organizzazione del disco.
In questo capitolo vengono esposti i criteri di memorizzazione delle informazioni e la struttura con cui questa memorizzazione viene mantenuta nella memoria di massa del nostro elaboratore. Viene introdotto il concetto di "file system" le cui caratteristiche, come vedremo, sono identiche a quelle del file system del DOS.

Capitolo 4 L'ambiente operativo di OS/2.
Questo capitolo tratta dell'interfaccia con cui OS/2 si presenta all'utilizzatore: la maschera video "program selector".
Vengono esposte le caratteristiche dei due diversi modi operativi, DOS e OS/2, e la funzionalità operativa, peraltro ancora simile a quella del DOS.

Capitolo 5 I comandi di OS/2: comandi "esterni" ed "interni"
Questo capitolo espone in ordine alfabetico tutti i comandi che accompagnano la versione 1.0 di OS/2 con molteplici esempi di utilizzo.

Capitolo 6 EDLIN: un programma per la gestione dei file di testo.
Questo capitolo è riservato all'esposizione dell'editor EDLIN, l'unico fornito con OS/2 e pertanto l'unico strumento con cui siamo in grado di scrivere e correggere file di testo.

Capitolo 7 OS/2 lavora per noi: i comandi per i file "batch".
In questo capitolo vengono esposti i comandi relativi ai soli programmi realizzati collezionando in un file di testo altri programmi; inoltre vengono introdotti i concetti di ambiente operativo e di gestione dell'Input/Output.

Capitolo 8 Come installare OS/2.
Questo capitolo tratta della procedura di installazione di OS/2, peraltro già automatizzata nei dischi di distribuzione.

Capitolo 9 Personalizzazione della configurazione di OS/2: CONFIG.SYS.
Questo capitolo espone in ordine alfabetico tutti i comandi di configurazione che l'utente può utilizzare per adattare OS/2 al proprio hardware e alle propie necessità operative; anche qui gli esempi sono numerosi.

Seguono tre appendici che espongono in forma tabellare le sequenze ANSI di controllo e il loro effetto, i layout delle tastiere delle varie nazioni, i diversi set di caratteri, codificati per poter essere configurati dall'utilizzatore, e una breve sintesi dei comandi di OS/2.
Infine, è disponibile un glossario dei termini utilizzati nel volume.

CAPITOLO UNO

Il sistema operativo: cos'è e cosa fa

La prima domanda a cui bisogna saper rispondere accingendosi alla lettura del presente volume è: "cos'è un sistema operativo?"
A tutti coloro che non vengono colti alla sprovvista da questa domanda consiglio senz'altro di passare senza indugi al capitolo successivo; a tutti gli altri è dedicato questo capitolo.
Prima di parlare del sistema operativo, cardine dell'Informatica, occorre premettere due concetti basilari: i nomi che li identificano sono oggi più che mai diffusi, tuttavia intendo ugualmente partire da essi affinché il discorso sia il più completo possibile: parlo naturalmente dell'Hardware e del Software.
Entrambi questi termini sono di origine anglosassone e identificano rispettivamente la parte visibile e tangibile dell'elaboratore (hard significa duro) e la parte invece non direttamente conoscibile (soft significa impalpabile).
I paragoni possibili per avvicinare questi concetti ad altri più familiari sono moltissimi e molto diffusi; dirò soltanto che, come ogni prodotto dell'uomo, anche il calcolatore si manifesta con una parte avente consistenza fisica (violino, propulsore di un'autoveicolo, edificio scolastico) e con la conoscenza necessaria per trarre da quest'ultima un utile per l'umanità (partitura musicale, ciclo Otto, didattica).

HARDWARE

Il calcolatore elettronico è composto da un certo numero di elementi, ciascuno specializzato in qualche funzione:

- Il video, o monitor, serve alla rappresentazione di dati (lettere, grafici, disegni, ecc.); costituisce quindi il mezzo tramite cui l'elaboratore mostra il risultato del lavoro compiuto: è quindi il dispositivo di uscita primario e si chiama comunemente STANDARD OUTPUT;
- La tastiera serve per comunicare all'elaboratore dati e istruzioni: è quindi il dispositivo di ingresso primario e si chiama comunemente STANDARD INPUT;
- L'unità centrale (CPU) assolve a tutte le funzioni inerenti la memorizzazione e l'elaborazione dei dati, ed è a sua volta suddivisa in altri componenti:
- La CPU (Central Process Unit) vera e propria, costituita da un componente, detto microprocessore e grande pochi centimetri quadrati, che è in grado di eseguire a velocità prodigiosa una serie di istruzioni elementari (le quattro operazioni e la possibilità di memorizzarne il risultato in una "cella" prescelta);
- La RAM (Random Access Memory), cioè l'insieme ordinato di tutte le "celle" in cui la CPU memorizza i dati e le istruzioni che le sono necessarie per il funzionamento (detta anche memoria dinamica o di lavoro);
- Le unità di lettura/scrittura a disco magnetico (Disk drive) che costituiscono il mezzo tramite cui è possibile memorizzare in maniera ordinata una grande quantità di dati e istruzioni (dette anche memoria di massa). I drive sono a loro volta suddivisi in due categorie: unità a disco rimuovibile e unità a disco fisso: le prime presentano un'apertura tramite cui è possibile sostituire il supporto magnetico (in anglosassone floppy disk, disco flessibile) a seconda delle necessità, a scapito però di una scarsa capacità di memorizzazione (da 360.000 a 1.400.000 caratteri) e una bassa velocità di accesso ai dati registrati sul supporto stesso; le seconde (in anglosassone hard disk, disco rigido) sono caratterizzate da una enorme capacità di memorizzazione (da 20 fino a 300 milioni di caratteri) e da una grande velocità di accesso ai dati registrati, fino a 10 volte superiore a quella delle unità a disco rimuovibile di cui chiaramente non condividono la possibilità della intercambiabilità del supporto magnetico.

L'attività dell'elaboratore consiste nell'esecuzione da parte della CPU di una serie di istruzioni che essa "reperisce" in alcune celle della memoria RAM, istruzioni fra cui frequentemente si trovano quelle responsabili del trasferimento del contenuto di parte della memoria di massa nella RAM (lettura); contenuto che, ad esempio, potrebbe essere costituito da dati o da altre istruzioni che, a loro volta, potrebbero essere quelle responsabili del trasferimento del contenuto della memoria RAM nella memoria di massa (scrittura) e così via: queste "istruzioni" e la loro struttura costituiscono il sistema operativo.

SOFTWARE

Il software è l'insieme di tutte le istruzioni che una CPU può eseguire sfruttando le periferiche dell'elaboratore (tastiera, video, memoria RAM, memoria di massa, stampanti, ecc.). Queste serie di istruzioni, chiamate con il nome generico di programmi, vengono poi classificate secondo gli scopi per cui i programmi stessi sono stati concepiti; esistono così programmi per fare altri programmi (linguaggi di programmazione), programmi applicativi specifici per una singola attività (fogli elettronici, data base managers, supporti per realizzazioni grafiche e acustiche, ecc.), programmi per l'automazione delle procedure d'ufficio (videoscrittura, correzione ortografica multilingue, impaginazione e stampa a corpi e stili differenziati, ecc.), programmi per la gestione delle procedure fiscali delle aziende (contabilità generale e IVA, gestione finanziaria, gestione cespiti, gestione magazzino, gestione bolle di accompagnamento merci, ecc.), programmi cosiddetti "ad hoc", realizzati cioè su commissione quindi in base a precise specifiche fornite dal committente.
Abbiamo visto che il sistema operativo è una serie di istruzioni: anch'esso quindi è un programma. Il sistema operativo è il programma che dialoga direttamente con le risorse hardware del computer e lo scopo della sua esistenza è quello di fornire agli altri programmi "un'interfaccia" semplice ed efficiente per l'utilizzo di dette risorse.
Questa definizione può essere chiarita con un esempio: la istruzioni che la CPU esegue sono estremamente elementari e il solo far comparire sul video il carattere del tasto che è stato premuto richiede una media di 10 istruzioni. Indipendentemente dal fatto che la CPU impiega un tempo impercettibile ad effettuare questa sequenza, se ognuno dei programmi che abbiamo elencato dovesse contenere tutte le istruzioni necessarie alla gestione delle periferiche (teniamo presente che l'esempio si riferisce al caso più semplice, cioè all'utilizzo del video e della tastiera; l'uso, ad esempio, dei dischi comporta centinaia di istruzioni per ogni operazione di lettura/scrittura), la stesura di detto programma diventerebbe laboriosissima e inoltre andrebbe di volta in volta adattata alle varie CPU e periferiche che il mercato mette a disposizione.
Quest'ultimo fatto, in particolare, impedirebbe la diffusione commerciale dei prodotti in quanto ogni programma diventerebbe del tipo "ad hoc" indipendentemente dal suo utilizzo, e questo perché diventerebbe dipendente dal tipo di hardware.
Il sistema operativo rappresenta la soluzione di questo problema: esso è un programma che trasforma una serie di istruzioni convenzionali nella sequenza di istruzioni elementari che quella particolare CPU con il suo contorno di periferiche è in grado di eseguire. In questo modo il solo programma che richiede di essere eseguito "ad hoc" per ogni macchina è, appunto, il sistema operativo: una volta cioè che un certo elaboratore sia dotato di quel sistema operativo, qualsiasi programma che faccia riferimento alle istruzioni convenzionali che detto sistema accetta, funzionerà su quell'hardware indipendentemente dalle sue caratteristiche.

Il sistema operativo, le cui caratteristiche verranno esposte nelle pagine seguenti, nasce dalla necessità di sfruttare appieno le caratteristiche di una CPU di recente realizzazione: il microprocessore 80286 realizzato dalla casa statunitense INTEL. Questo componente possiede le caratteristiche necessarie alla realizzazione di un sistema operativo multiprocesso (traduzione semiletterale dell'anglosassone multitasking), di un sistema operativo cioè che sia in grado di eseguire contemporaneamente più programmi sulla stessa unità.
Questo microprocessore è la naturale evoluzione del 8086, della stessa casa costruttrice, che equipaggia i Personal Computer IBM e tutta quella schiera di elaboratori elettronici, detti "compatibili", che hanno invaso il mercato dopo la comparsa di detto calcolatore. Il "fenomeno" Personal Computer che, come dice il nome, è caratterizzato dalle ridotte dimensioni e soprattutto dalla possibilità di venir usato in qualsiasi condizione, ha svincolato il mondo dell'informatica dalle grandi sale climatizzate dei centri EDP (sigla che sta per Electronic Data Processing, cioè elaborazione di dati e informazioni realizzata con l'ausilio dell'elaboratore elettronico), avvicinando il calcolatore al pubblico. La caratteristica che ha permesso questa affermazione è proprio il sistema operativo fornito con queste macchine che, avendo raggiunto una diffusione mondiale, ha favorito la creazione di una sorta di backgroud conoscitivo che è poi il vero movente della penetrazione di questi prodotti.
E proprio alla popolarità raggiunta dal sistema operativo in questione è dovuta la nascita del naturale successore che ci accingiamo ad apprendere.

I sistemi operativi per personal computer: DOS e OS/2

MS-DOS è il nome che la Microsoft (le due lettere MS rappresentano appunto la "sigla" di questa ditta statunitense) ha scelto per il suo Disk Operating System (la versione adottata dalla IBM è sostanzialmente uguale ma porta il nome PC-DOS) che tanto successo ha avuto e tanta ricchezza ha portato ai suoi ideatori. Eppure, da un punto di vista prettamente tecnico, questo sistema operativo mostra tantissimi limiti rispetto a quelle che sarebbero le potenzialità del microprocessore su cui è stato sviluppato (ad esempio, esiste un limite alla quantità di memoria RAM che questo sistema può utilizzare, indipendentemente dalla capacità del microprocessore che è molto maggiore) e, a mano a mano che gli utenti di detto sistema crescevano in numero e capacità, questi limiti si sono fatti sentire in maniera sempre più pesante.

MS-DOS e PC-DOS, equivalenti per le caratteristiche richieste nella nostra esposizione, saranno, da qui in avanti, individuati univocamente dal termine DOS.

Due sono le mancanze che tutti rimproverano al DOS: la più importante è senz'altro l'impossibilità da parte del sistema operativo di superare il fatidico muro di 640 Kbyte di espansione massima della memoria RAM (Kbyte = chilobyte = 1024 byte: un byte è lo spazio minimo necessario al calcolatore per memorizzare un carattere, sia esso numerico o alfabetico e, poiché tutti i dati e le istruzioni necessarie all'elaboratore durante l'esecuzione di un programma devono risiedere nella memoria, la capacità di quest'ultima diventa una discriminante sempre più significativa a mano a mano che i programmi aumentano in complessità e prestazioni). L'altra grande mancanza del DOS è l'impossibilità di eseguire più programmi contemporaneamente: occorre questa volta considerare che questa

limitazione è imposta dalle caratteristiche del microprocessore che sta alla base della progettazione dei Personal Computer (l'8086 della statunitense INTEL), il quale non permette di "proteggere" la memoria che contiene dati ed istruzioni del primo programma da eventuali "sovrapposizioni" che potrebbero venir causate da un secondo programma in esecuzione nello stesso momento.
Il DOS aveva anche un limite concernente la dimensione massima del supporto di memoria di massa; tuttavia questo problema, peraltro già superato da fornitori di strumenti software che hanno permesso di utilizzare hard disk di gran lunga più capaci dei soli 32 Mbyte (Megabyte = 1 milione di byte) del DOS, oggi non esiste più in quanto dalla versione 3.3 di questo sistema operativo è possibile appunto utilizzare dischi di capacità molto maggiore.
Sollecitata dal mercato e, probabilmente, dal suo grande partner IBM, la stessa casa ha presentato Operating System/2: le due doti fondamentali del nuovo sistema operativo, non a caso, sono proprio la possibilità di sfruttare tutta la memoria RAM che il microprocessore è in grado di gestire e il multitask, la capacità cioè di eseguire più programmi contemporaneamente.
La possibilità per un sistema operativo di consentire il multitask è subordinata alla caratteristica che la CPU, su cui detto sistema viene sviluppato, ha di permettere la "protezione" della memoria, in altri termini dev'essere in qualche modo possibile delimitare i confini della memoria RAM dedicata a ciascun programma in esecuzione, specialmente quella occupata dal sistema operativo, al fine di evitare che una incidentale sovrapposizione arresti il sistema. Il microprocessore che permette questa metodologia operativa è, in questo caso, l'80286 prodotto dalla INTEL: questo componente equipaggia tutti i Personal Computer dell'ultima generazione, dai Personal System/2 della IBM (ad esclusione del modello 30) agli Olivetti M280 e M380, Compaq286 e Compaq386, Unisys Power Workstation/2 e, in generale, tutti i modelli di elaboratore compatibili con il Personal Computer IBM PC/AT.
Questo microprocessore può essere utilizzato secondo due diverse modalità: in modo "reale" e in modo "protetto". Molti calcolatori montano già da diverso tempo questo componente ma, poiché il solo sistema operativo disponibile era il DOS, utilizzavano l'80286 in modo reale: così facendo la memoria utilizzabile rimane di 640 Kbyte e non è possibile avere il multitask. Il passaggio da una modalità all'altra è totalmente delegato al sistema operativo che viene caricato sull'elaboratore al momento dell'accensione: è sufficiente infatti caricare OS/2 sui suddetti calcolatori e il microprocessore 80286 dicui essi sono dotati funzionerà in modo protetto, permettendo l'utilizzo dell'ammontare massimo di memoria utilizzabile da questo componente, pari a 16 Mbyte, e l'esecuzione contemporanea di più processi.
Operating System/2 viene in effetti distribuito in due diverse versioni: una per i Personal Computer del tipo IBM PC/AT e compatibili (elaboratori distribuiti dal 1984 e che finora hanno utilizzato l'80286 in modo reale con il DOS), e una per le nuove famiglie di personal che hanno la caratteristica peculiare di montare i drive di nuovo formato (3,5 pollici, alta capacità).

La complessità di OS/2 fa sì che le dimensioni del sistema operativo siano di gran lunga maggiori di quelle del DOS: la quantità minima di memoria necessaria per utilizzare OS/2 è di 1,5Mbyte. Va comunque rilevato che OS/2 possiede la capacità di passare rapidamente da RAM a disco parti di programma nei momenti in cui questi sono inattivi (questa metodologia operativa è detta "swapping"); questo permette, una volta garantito l'ammontare minimo di memoria, di eseguire un numero di processi per i quali la somma algebrica della memoria totale richiesta risulterebbe maggiore di quella disponibile.
Una importante caratteristica di questo sistema operativo è la disponibilità di una zona di memoria, di grandezza variabile e di limite massimo uguale a 640 Kbyte, in cui il programma viene eseguito in modo reale; vale a dire esiste la possibilità di eseguire sotto OS/2 tutte le applicazioni standard scritte per il DOS (occorre tenere presente che per avere l'estensione massima di memoria per il modo reale occorre aumentare corrispondentemente la memoria RAM fisica del calcolatore ed inoltre che può essere attivato un solo processo con questa caratteristica, indipendentemente dal numero di altri processi attivi in modo protetto). A questo proposito, la Microsoft ha definito un sottoinsieme di "chiamate di sistema", detto "Family API (Application Program Interface)", che permette di creare programmi che possono essere eseguiti indipendentemente nel modo reale o protetto di OS/2. In definitiva, possiamo individuare tre tipologie di applicazioni che il nuovo sistema permette:

- quelle classiche, create per il DOS, che in OS/2 vengono eseguite nel modo reale (se la memoria è sufficiente);
- quelle realizzate secondo lo standard Family API, che vengono eseguite in entrambi i modi in OS/2, e che possono quindi ancora essere eseguite in DOS (questo perché il modo reale di OS/2 è esattamente equivalente al DOS per cui, così come un programma per il DOS funziona in OS/2 in modo reale, allo stesso modo un programma creato per il modo reale di OS/2 funziona in DOS);
- quelle realizzate facendo un uso esteso delle chiamate di sistema che definiscono l'insieme API, che vengono eseguite solo nel modo protetto di OS/2.

Ad ogni processo OS/2 associa almeno una maschera video: questo prende il nome di "screen group". Il passaggio da uno screen group ad un altro è comandato dall'utente tramite la combinazione dei due tasti ALT e ESC; i tasti CTRL e ESC, invece, presentano uno screen group particolare, detto "program selector screen" o "process manager screen": tramite questa maschera video è possibile osservare i processi in esecuzione e attivarne di nuovi. È importante notare che, mentre i vari processi in esecuzione in modo protetto rimangono in esecuzione anche se il loro screen group non è quello presente sul monitor del calcolatore, il processo che eventualmente si trova in esecuzione in modo reale è attivo solo quando è effettivamente padrone del monitor e della tastiera; è quindi inattivo quando un

altro screen group o il program selector screen occupano il monitor.
La versione di OS/2 attualmente in distribuzione è denominata Versione 1.0 ed è quella che viene presentata in questo volume: la differenza fondamentale fra questa e la Versione 1.1, che è stata annunciata per la fine del 1988, sta nel fatto che quest'ultima sostituirà al classico "prompt" a cui siamo abituati per eseguire i programmi (il prompt, o invito, è il segno che si presenta sulla sinistra della linea del monitor di un elaboratore che, una volta acceso e caricato il sistema operativo, sia in attesa di istruzioni; per il DOS e OS/2 Versione 1.0 questo prompt si presenta con C> o A>, a seconda che l'elaboratore sia dotato o meno di un hard disk) un programma, il cui nome è "Presentation Manager", che, in una forma simile a quella del prodotto Windows, offre all'utente un'interfaccia molto evoluta per l'esecuzione dei comandi del sistema operativo.
Terminiamo questo capitolo con una breve sintesi delle caratteristiche di OS/2 e dei requisiti minimi per utilizzarlo:

CPU	Intel 80286 e 80386 (Utilizzate in modo protetto, cioè multitask)
Architettura calcolatore	IBM XT286, IBM PC/AT e compatibili IBM PS/2 e compatibili
Memoria RAM	1,5 Mbyte (2Mbyte per il modo reale)
Memoria di massa	Hard Disk da 10Mbyte minimo
Versione	1.0
Spazio occupato su disco	minore di 3 Mbyte

CAPITOLO TRE

L'organizzazione del disco

Il FILE: LA MEMORIZZAZIONE DI "OGGETTI" DIVERSI SUL DISCO

Il risultato dell'utilizzo di un elaboratore elettronico può assumere la forma di pagina stampata, può essere un risultato temporaneo che appare sul monitor o, ed è il caso più generale, può venire memorizzato per un uso futuro. I programmi che noi utilizziamo, devono anch'essi essere memorizzati in modo permanente, al fine di permetterci di ripetere le medesime elaborazioni ogni qual volta ne abbiamo la necessità.
La memoria di massa di un calcolatore permette di memorizzare dati estemporanei, programmi applicativi, archivi in continuo aggiornamento e, naturalmente, lo stesso sistema operativo. A tutti questi diversi oggetti viene assegnato un nome e tutte le operazioni di memorizzazione e di successivo reperimento del loro contenuto avvengono tramite questo nome. Inoltre, poiché, una volta che sia stato assegnato loro un nome, i diversi oggetti ci appaiono semplicemente come un elenco, è invalso l'uso della parola anglosassone "file" per identificare un qualsiasi oggetto memorizzato sul disco di un elaboratore, lasciando poi all'utilizzatore il compito (e la libertà) di associare nomi particolari a contenuti particolari.
Esistono evidentemente molti comandi per scorrere questo elenco di nomi e per ottenerlo secondo diversi ordinamenti (alfabetico, per lunghezza del file, per data di ultima modifica del contenuto, ecc.), tuttavia una memorizzazione priva di qualsiasi organizzazione si tradurrebbe inevitabilmente in elenchi lunghissimi e dal contenuto talmente eterogeneo da rendere difficoltosa l'individuazione del file desiderato. Per evitare questo inconveniente e per fornire una struttura organica agli oggetti che memorizziamo, la memoria di massa del nostro elaboratore è organizzata in un "file system". Il file system ci permette di catalogare i nostri file

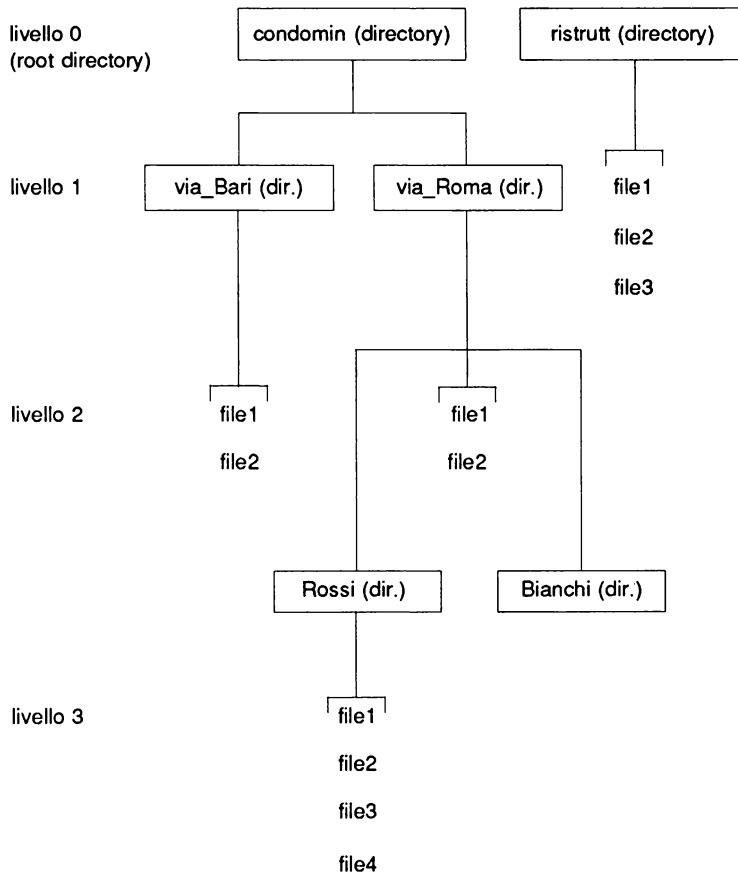
tramite il loro nome, e secondo il loro contenuto, in una struttura gerarchica che vede gruppi omogenei di file, posti ad un determinato livello nella struttura, riepilogati con un unico nome che compare al livello superiore della struttura.

IL "FILE SYSTEM": L'ORGANIZZAZIONE GERARCHICA DEI FILE

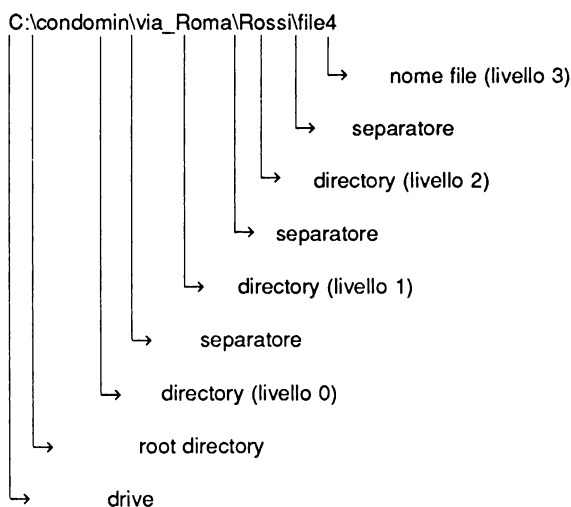
Facciamo un esempio: supponiamo che il nostro lavoro consista nel seguire le pratiche di gestione di un certo numero di condominii con più affittuari ciascuno e di seguire la pratica per i lavori di ristrutturazione di un immobile. Innanzitutto, abbiamo una prima grossa divisione: avremo sicuramente due diverse cartelle (o due diversi schedari) per le pratiche relative alle due attività, avremo scritto sopra ciascuna di esse una sigla per identificarne il contenuto e apriremo l'una o l'altra a seconda della necessità del momento. All'interno della cartella (o dello schedario) relativa alle pratiche di gestione dei condominii avremo probabilmente una cartella per ogni singolo condominio intestata con l'indirizzo dello stabile e, anche questa volta, apriremo quella dello stabile di cui ci stiamo occupando in quel momento. Nulla vieta che all'interno di questa cartella, oltre ai documenti riguardanti l'intero stabile, vi sia una ulteriore cartella per ogni singolo condomino, intestata con il suo nome, che noi apriremo ogni qual volta dovremo occuparci di problematiche che lo riguardano.
I diversi livelli della struttura del file system equivalgono alle nostre cartelle: quando sono chiuse noi vediamo solo le due diverse sigle ma non il rispettivo contenuto; aprendone una non vediamo più la sigla ma vediamo tutte le cartelle e i rispettivi indirizzi degli stabili: l'informazione è ancora sommaria; solo aprendo la cartella di un condominio possiamo prendere visione del contenuto che, oltre ai documenti dello stabile che equivalgono ai nostri file, può, ancora una volta, essere composto da cartelle, individuate dal nome del condomino, che occorre aprire per poter prendere visione dei documenti (i file) del livello più basso.
Se i documenti del nostro esempio equivalgono ai file nel file system, qual è il parallelo per i nomi o le sigle delle cartelle? La risposta a questa domanda è la chiave del file system e si chiama "directory": l'elenco dei file di un determinato livello (directory, appunto) mostra i nomi di tutti i file contenuti in quel livello e i nomi delle directory visibili da quel livello (nel nostro esempio, l'elenco di tutti i documenti di uno stabile e delle cartelle di tutti i condomini); scendendo di un livello ("entrando" in una delle directory visibili) l'elenco conterrà soltanto i file di quel livello (nel nostro esempio, l'elenco di tutti i documenti riguardanti un preciso condomino).
Ricapitolando, il file system è una struttura gerarchica organizzata secondo livelli, in ciascuno dei quali possono trovarsi indipendentemente file e directory: la presenza in un livello di una directory determina l'esistenza di un livello più basso in cui possono ancora trovarsi indipendentemente file e directory: risulta chiaro che questo schema può ripetersi verso il basso senza limiti. Esiste invece un livello

superiore da cui si origina questa struttura: questo livello, detto "root directory", cioè radice, è associato al drive e quindi esiste per ogni drive di cui il sistema è dotato. A questo proposito è bene precisare che, mentre esiste sempre una root directory, la realizzazione della struttura suesposta è comunque affidata all'utilizzatore: il sistema operativo si limita a mettere a disposizione i comandi per la sua implementazione.

Il file system viene spesso paragonato ad un albero rovesciato: la radice è in effetti il livello più alto della struttura da cui si dipartono tutti i rami (non a caso la directory da cui si dipartono tutte le altre è detta root). Un ramo può poi dare origine ad altri rami, portare solo foglie o portare foglie pur biforcandosi in altri rami: anche in questo caso il parallelo con le directory del file system è immediato (i file sono le foglie del nostro albero).

Il disegno seguente rappresenta lo schema finora illustrato (i criteri di definizione dei nomi sono spiegati più avanti):

Il livello 0 o root directory esiste sempre; il livello 1 esiste perché nella root directory figurano 2 directory (evidentemente create da noi). In una di queste due (condominii) figurano altre due directory: ciò significa che esiste il livello successivo (livello 2 in figura). Qui, oltre ad un certo numero di file, figurano altre directory: esiste quindi un livello ulteriore (livello 3 in figura). Poiché in quest'ultimo livello figurano solo file, la struttura gerarchica del file system di questo disco ipotetico si ferma qui: è evidente che in qualsiasi momento possiamo aggiungere e togliere file e directory a qualsiasi livello modificando secondo le nostre necessità la suddetta struttura.

IL PATHNAME: LA LOCALIZZAZIONE DI UN FILE ALL'INTERNO DEL FILE SYSTEM

Esiste una simbologia per individuare gli elementi di un file system: ogni file memorizzato nella memoria di massa del computer può essere identificato dal "percorso" che, partendo dalla root directory di un particolare drive, attraversa i vari livelli per arrivare appunto al file. Innanzitutto esiste una simbologia per identificare i drive: A e B identificano rispettivamente il primo e il secondo drive per i floppy disk; C e D identificano rispettivamente il primo e il secondo drive hard disk. La simbologia di identificazione di un elemento del file system prevede che il percorso inizi con la lettera che individua il drive contenente il file system in oggetto seguita da ":", poi dal simbolo "\" che indica la root directory, a sua volta seguito dal nome della directory in cui si vuole "entrare" per passare al livello inferiore, seguito ancora dal simbolo "\", questa volta come separatore, ripetendo poi questa operazione fino al raggiungimento del file desiderato che rappresenta l'elemento conclusivo di detto percorso. Supponendo che il file system dell'esempio precedente sia quello del drive C, la simbologia che identifica il "file4" del terzo livello è:

Quando, nella simbologia di identificazione di un file, si dettaglia l'intero percorso a partire dalla root directory,o addirittura dal disco che identifica il file system di riferimento, si parla di "pathname assoluto" di quel particolare file (il termine anglosassone path significa appunto percorso). L'aggettivazione assoluto è motivata dalla evidente ragione che la simbologia or ora illustrata identifica in modo inequivocabile il file indicato; va comunque rilevato che spesso la nozione di pathname assoluto si applica anche quando la lettera identificativa del drive non è specificata (cioè il pathname inizia con "\"); questo ci porta alla definizione di drive e directory "correnti": ad ogni istante, all'utilizzatore di un personal computer sono associati un disco e una directory a cui, salvo diversa specifica da parte dell'utilizzatore, fanno riferimento tutti i comandi che interagiscono con il file system. In altre parole, l'utente è situato all'interno di un livello della struttura gerarchica e i suoi comandi possono utilizzare una simbologia ridotta perché danno per scontata una parte di percorso. Se, ad esempio, all'utilizzatore è associata la directory "via_Roma" del drive C, l'identificazione dello stesso file dell'esempio precedente può ridursi a:

```
Rossi\file4
```

così come l'identificazione di "file1", che si trova al medesimo livello di quello in cui si trova l'utilizzatore, si riduce semplicemente al nome (anziché al suo pathname assoluto che sarebbe C:\condomin\via_Roma\file1). I percorsi scritti a partire dalla directory del drive associata all'utilizzatore vengono detti "pathname relativi". Esiste una particolare simbologia che permette, per questi ultimi, di "risalire" i livelli per identificare file che si trovano magari allo stesso livello, ma catalogati sotto un'altra directory: supponiamo ancora che al nostro utilizzatore sia associata la directory "via_Roma" del drive C dell'esempio precedente ma che il file da raggiungere sia "file2" posto sotto la directory "via_Bari". Questo file non può essere chiaramente raggiunto scendendo lungo i livelli della nostra directory corrente, per cui parrebbe che l'unica via per identificare questo file sia di darne il pathname assoluto, e cioè:

```
C:\condomin\via_Bari\file2
```

Se però avessimo modo di specificare il percorso risalendo al livello superiore (directory condomin) e scendendo poi lungo la directory che contiene "file2", daremmo una identificazione più sintetica e funzione della directory a noi associata. La simbologia "..", chiamata "directory padre" dall'anglosassone "parent directory", assolve questa funzione, per cui il percorso per raggiungere "file2" dalla directory a noi associata diventa:

```
..\via_Bari\file2
```

che risulta evidentemente più sintetica. L'ultima simbologia che resta da intro-

durre è quella che identifica la directory corrente: esistono comandi in cui è necessario specificarla e, quando occorre, lo si fa con il simbolo ".".
Il file system è comune a DOS e OS/2: quanto detto finora si applica ad entrambi i sistemi operativi con l'evidente vantaggio che chi possiede già un Personal Computer dotato di DOS può passare a OS/2 senza dover salvare né alterare il contenuto dei suoi dischi, ad eccezione dei file che contengono il sistema operativo.

I NOMI DEI FILE E DELLE DIRECTORY: CRITERI PER LA LORO DEFINIZIONE

Esiste anche una simbologia per caratterizzare il contenuto di un file. Occorre precisare innanzitutto che il nome di un file è composto da due parti: la prima parte è il nome vero e proprio, le seconda parte è detta estensione e costituisce un metodo per classificare i file per contenuto. Il nome è composto da un massimo di 8 caratteri qualsiasi, esclusi:

```
*?/\.,;:+=<>[]()&
```

tuttavia è consigliabile usare solo lettere, numeri e il segno "_", perché potrebbero insorgere difficoltà se dovessimo trasferire questi file su sistemi operativi diversi da OS/2 e che non permettono caratteri particolari). L'estensione, se presente, inizia con ".", seguito da un massimo di 3 caratteri, secondo le regole precedentemente esposte. Per i nomi delle directory vale lo stesso criterio usato per i file: sarà OS/2 a indicarci quand'è che un nome individua una directory e quand'è che individua un file. Nel nome di un file o directory è assolutamente vietato inserire spazi perché, essendo lo spazio trattato come carattere di separazione, quello che per noi è un nome contenente uno spazio, viene considerato dal sistema operativo come una coppia distinta di nomi. I nomi di file e directory utilizzati nell'esempio precedente sono stati costruiti secondo i criteri esposti.

RUOLO DEI CARATTERI "?" E "*" NELLA LOCALIZZAZIONE DIINSIEMI DI FILE

Per ciò che riguarda la localizzazione dei file, esistono dei caratteri che, posti all'interno del nome di un file, assumono un significato particolare. Questi caratteri, in anglosassone "wildcard", sono "?" e "*" e vengono comunemente indicati come "jolly". In effetti, il loro scopo è di sostituire, secondo un criterio posizionale, una serie di caratteri nel nome di un file, per permettere che, con quell'unico nome, si possa individuarne un gruppo.
Se, all'interno del nome di un file, viene inserito il "?", nella posizione da esso occupata può esservi qualsiasi altro carattere, vale a dire che i tre file di nome FILE10, FILE20 e FILE30 sono tutti identificati dal nome FILE?0, mentre non lo è il file di nome FILE0. Il carattere "?", pertanto, sostituisce uno e un solo carattere;

naturalmente in un nome esso può comparire più volte.
Se, all'interno del nome di un file, viene inserito il "*", nella posizione da esso occupata può esservi un numero di caratteri a piacere: nell'esempio appena esposto, il nome FILE*0 identifica sia FILE0 che FILE10, FILE20, FILE30 ed ogni altro eventuale file il cui nome inizi con FILE e termini con 0, indipendentemente dal numero e dal tipo degli altri caratteri. Il carattere "*", pertanto, sostituisce da 0 al massimo numero di caratteri permessi.
Nome ed estensione sono considerate dai jolly come parti distinte; in altri termini, il nome FILE*0 non identifica un file di nome, ad esempio, FILE50.DAT: occorre infatti ripetere l'operazione di sostituzione sull'estensione secondo gli stessi criteri esposti per i nomi: un nome adatto a quest'ultimo esempio è FILE*0.* oppure, per limitare l'insieme dei file identificati, FILE*0.D??.
L'utilizzo dei caratteri jolly si rivelerà particolarmente utile per tutti quei casi in cui occorre identificare un gran numero di file: se si è usato l'accorgimento di utilizzare per il loro nome una radice comune o un'estensione comune, i caratteri jolly ci permetteranno di trattarli tutti, ad esempio in un'operazione di copia, in un unico comando.

CON, NUL, PRN, AUX, LPT, COM: I NOMI DEI FILE "DEVICE"

Alcuni nomi, benché utilizzati dai vari comandi come nomi di file, non figurano in alcuna directory di alcun disco: la loro funzione è quella di fornire un mezzo per caratterizzare i vari dispositivi, dall'anglosassone "device", di cui il calcolatore può essere dotato. Questi nomi convenzionali hanno la medesima funzione che le lettere A, B, C e D hanno per i drive a dischi: essi designano cioè un dispositivo fisico le cui risorse possono venir utilizzate tramite la specifica del nome, proprio come il pathname assoluto del nome del drive individua univocamente un file. Proseguendo nell'esposizione del sistema operativo incontreremo spesso questi nomi, per ora elenchiamo i loro significati:

CON	Console device (designa l'insieme tastiera e monitor)
NUL	Null device (designa il device cui dirigere tutto ciò che non deve apparire sul monitor)
PRN	Printer device (designa il device generico cui è connessa la stampante)
AUX	Auxiliar device (designa un generico device ausiliario)
LPT	Line parallel printer device (designa, seguito da un numero, una delle possibili stampanti parallele connesse al computer)
COM	Serial line device (designa, seguito da un numero, una delle linee di comunicazione seriale di cui è dotato il computer)

I termini parallelo e seriale si riferiscono al metodo con cui i dati escono o entrano

nell'elaboratore. I dati sono costituiti da un certo numero di bit (l'unità elementare della logica di elaborazione, corrispondente ai due valori binari 0 e 1 che corrispondono a loro volta alla presenza o assenza del segnale elettrico) i quali, considerati a gruppi di otto (valore che definisce il byte), costituiscono una "parola" per il calcolatore. Affinché l'elaboratore possa scambiare dati con i vari dispositivi (ad esempio, inviare ad una stampante il testo da stampare), occorre che il metodo con cui il calcolatore invia o riceve queste parole sia lo stesso con cui detto dispositivo le invia o le riceve a sua volta.

Senza entrare troppo nel dettaglio, possiamo paragonare questi due metodi ad una classe di scolari che marcia in un cortile: se ogni scolaro corrisponde ad un bit, quando gli scolari marciano in fila per otto ogni riga della fila porta una parola (comunicazione parallela); se marciano in fila per uno, per ottenere una parola occorre sapere da quale scolaro inizia (quindi occorre un segnale particolare di inizio parola), quindi contare otto righe della fila e così via (comunicazione seriale). Il vantaggio principale del metodo parallelo è una maggior velocità di comunicazione in quanto viene trasferita una intera parola per volta; lo svantaggio è che la distanza fra computer e dispositivo dev'essere molto ridotta, dell'ordine della decina di metri. Al contrario, la comunicazione seriale può coprire distanze dell'ordine del centinaio di metri; tuttavia il trasferimento di un bit per volta, che costringe la ricomposizione della parola una volta a destinazione, comporta una riduzione della velocità di comunicazione.

CAPITOLO QUATTRO

L'ambiente operativo di OS/2

IL PROGRAM SELECTOR: LA GESTIONE MULTITASK

Completata l'esposizione delle caratteristiche strutturali di OS/2 e dell'elaboratore che lo supporta, siamo pronti per iniziare l'approccio con il sistema operativo. Supponiamo di avere a disposizione un Personal Computer su cui sia installato OS/2 (da qui in avanti il computer sarebbe meglio averlo, piuttosto che supporlo; in questo caso, troveremo una guida all'installazione di OS/2 nel capitolo 8): quando lo accendiamo, sullo schermo appare una maschera video riportata in figura 4.1.
Questo programma ci permette di scegliere l'ambiente in cui vogliamo operare; in altre parole, possiamo scegliere a quale programma associare il monitor e la tastiera del nostro elaboratore. Naturalmente, questa scelta non èassolutamente definitiva: in qualsiasi momento, durante l'esecuzione del programma selezionato, possiamo far riapparire la maschera del program selector e attivare un altro programma.
La struttura del program selector è molto efficace anche grazie all'interfaccia di tipo grafico che ci permette di interagire con il sistema operativo anche se lo conosciamo ancora molto poco.
Innanzitutto, lo schermo è diviso in tre parti che presiedono tre diverse funzioni: la riga superiore a colore di fondo invertito presiede alle funzioni di aggiornamento della maschera stessa, oltre a fornire istruzioni per usufruire della funzione "guida" F1; la colonna di sinistra elenca i nomi dei programmi che possono essere attivati dal program selector (è quella che viene aggiornata dalla funzione appena esposta) e infine, la colonna di destra elenca i programmi che sono in esecuzione e permette di passare dall'uno all'altro.

```
Update                                                    | F1=Help
                         Program Selector

Use← or → to move between Start a Program and Switch to a Running Program.
Use ↑ or ↓ to move select bar, then press Enter. Press F10 to Update lists.

 Start a Program                        Switch to a Running Program

 •* OS/2 Command Prompt *               •MS-DOS Command Prompt
 •Introducing OS/2
```

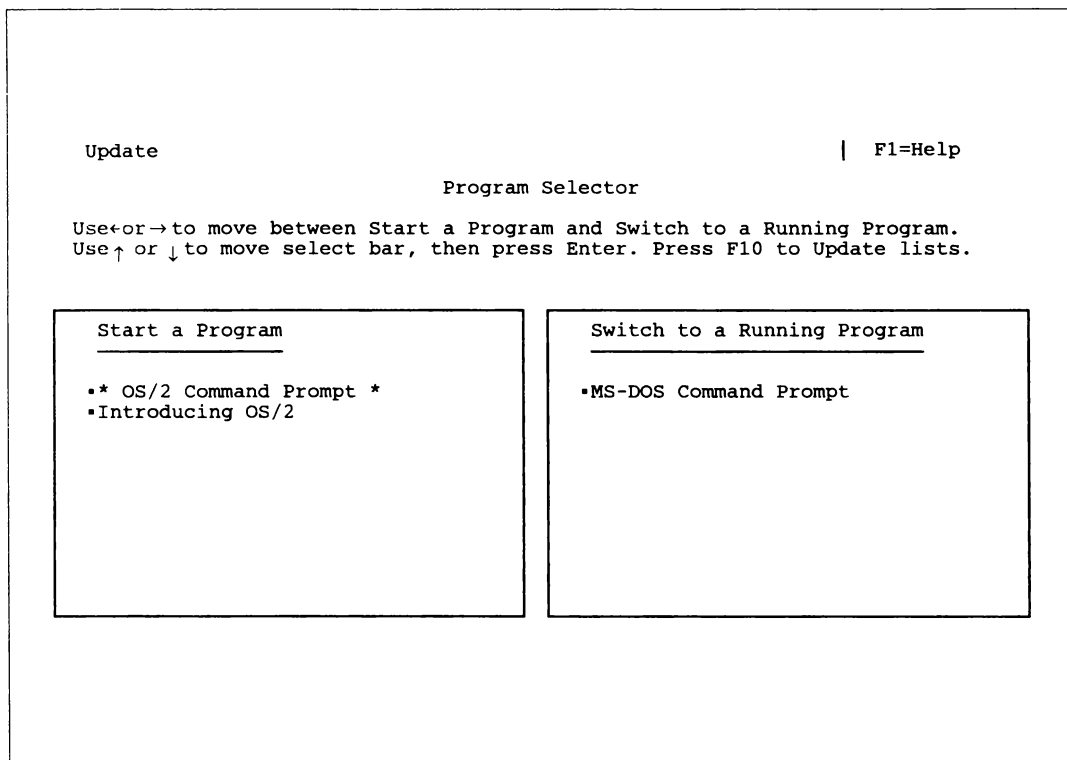



Figura 4.1

L'interazione avviene tramite l'uso delle frecce direzionali (o del mouse): il colore di fondo invertito si sposta da un nome all'altro nella stessa colonna o nelle due colonne in accordo ai tasti direzionali che noi premiamo; premendo il tasto ENTER (o INVIO o RETURN o CR o "↵": la dicitura riportata sul tasto dipende totalmente dalla fantasia dei costruttori di elaboratori) si effettua la selezione del programma il cui nome compare sul fondo inverso. Se al momento della selezione ci trovavamo nella colonna di destra, la maschera del program selector scompare, lasciando il posto al programma già avviato, a cui vengono associati video e tastiera; se invece eravamo nella colonna di sinistra, il programma viene avviato e la colonna di destra del program selector viene aggiornata prima di proseguire come nel caso precedente.
Anche in questo caso, un parallelo può aiutarci a comprendere più facilmente la funzionalità di questa sorta di smistamento: in effetti, proprio come da un unico binario ferroviario, approssimandosi ad una stazione di testa, si dipartono i vari tronchi che vanno alle pensiline, così dal program selector screen noi "imbocchiamo" il programma che abbiamo scelto. E proprio come il treno che ha imboccato il binario che porta ad una pensilina può essere fatto ritornare allo scambio per cambiare tronco e pensilina, anche noi possiamo in qualsiasi momento ritornare

fino al program selector per scegliere un altro programma.
La scelta di passare da un programma all'altro porta ad associare monitor e tastiera a quel particolare programma; in altri termini, passare da un programma ad un altro non ha altro effetto apparente che cambiare la maschera video: questa associazione programma-video prende il nome di "Screen Group". È necessario soffermarsi su questo concetto: dire che esiste uno screen group per ogni programma attivato tramite il program selector non significa necessariamente che per ogni programma in esecuzione esista un differente screen group: se, avviato un programma dalla maschera del program selector (ad esempio, LOTUS 123 che permette a sua volta di richiamare un altro programma), ne avviamo un secondo senza ripassare per il program selector, questi due programmi fanno parte del medesimo screen group. Come si vede, il concetto di screen group è più vicino a quello di sessione: non a caso, infatti, un altro nome che viene frequentemente attribuito al program selector è "Session Manager".

PROGRAMMI DOS E OS/2: COME GESTIRLI TRAMITE IL PROGRAM SELECTOR

Dopo l'installazione, nella colonna di destra, dove sono elencati i programmi in esecuzione, si trova la voce "DOS Command Prompt": se, usando i tasti di direzione, posizioniamo il blocco indicatore a fondo inverso su quest'ultima e premiamo ENTER, il video si pulisce e compare un prompt (il segnale con cui il calcolatore ci comunica che è pronto a lavorare per noi e da quale drive partire) così formato (supponendo che OS/2 sia stato installato sul disco rigido C):

```
C>
```

La sessione che abbiamo così iniziato è quella che emula l'ambiente DOS, già familiare a molti di noi (il microprocessore opera in modo reale, cioè il modo operativo finora usato dai Personal Computer AT e compatibili che funzionano in DOS).
In queste condizioni, la quasi totalità dei programmi scritti per DOS possono venir eseguiti, persino quelli che fanno uso di funzioni non documentate (come ad esempio SideKick), mentre avranno difficoltà a funzionare quei programmi che accedono direttamente ai componenti hardware (esempio il controller della memoria RAM) o che, per problemi di sincronizzazione con altri dispositivi esterni (drive con diverse prestazioni) riprogrammano il controller dei drive dei dischi. I comuni applicativi (praticamente tutti i programmi di trattamento testi, i fogli elettronici, database, ecc.) funzionano senza problemi all'interno di questo screen group (detto "compatibility box"), anzi, molti di essi funzionano anche fuori dal compatibility box. I programmi che possono risentire del diverso, anche se molto ben realizzato, ambiente sono alcuni "copiatori" e i programmi che per offrire prestazioni grafiche elevate fanno uso di dispositivi hardware aggiuntivi che sono in conflitto con OS/2.

Naturalmente, in presenza di dubbi, non esiste altro metodo che la prova per verificare se questi problemi esistono o meno.
Per ritornare al program selector, occorre premere simultaneamente i tasti CTRL e ESC: questa operazione può essere eseguita in qualsiasi momento, anche se un programma era in corso di esecuzione; infatti, quel programma resterà sospeso al punto in cui era arrivato e riprenderà la sua attività solo quando verrà di nuovo selezionato il suo screen group.
Se adesso, sempre con i tasti direzionali, selezioniamo "OS/2 Command Prompt" nella colonna di sinistra (l'intestazione "Start a Program" significa, appunto, avvia un programma), ancora una volta lo schermo si pulirà e comparirà un prompt come questo:

```
[C:\]
```

Questa volta la sessione che abbiamo iniziato è in modo protetto e, se durante l'esecuzione del programma, noi ritorniamo al program selector, detta esecuzione non si arresterà se non alla fine del programma. Finché ciò non avviene, la colonna di destra della maschera del program selector riporta due voci (i nomi dei due screen group avviati), tuttavia la voce OS/2 Command Prompt rimane anche nella colonna di sinistra. In effetti è possibile siaritornare alla sessione appena avviata selezionandola dalla colonna di destra, che attivare una nuova sessione in modo protetto, selezionando OS/2 Command Prompt dalla colonna di sinistra: non esiste cioè limite, fatte salve le risorse del calcolatore, al numero di screen group che possono essere attivi contemporaneamente in modo protetto. Il multitask effettivo è quindi limitato agli screen group avviati in modo protetto; infatti l'esecuzione di ciascun programma attivato in questo modo prosegue anche se non ha il controllo del monitor, contrariamente all'unico screen group che permette la compatibilità con i programmi esistenti in DOS, i quali sono attivi solo quando controllano il monitor.
Per passare da un programma a un altro (o meglio, da una sessione a un'altra), oltre all'ovvia soluzione di richiamare la maschera del program selector e selezionare sulla colonna di destra il nome dello screen group scelto (l'intestazione "Switch to a running program" significa, appunto, passare da un programma in esecuzione ad un altro), esiste la possibilità di passare da una sessione all'altra in modo continuo, come attraversare una serie di stanze poste una dopo l'altra lungo un corridoio circolare: per farlo, è sufficiente premere contemporaneamente i tasti ALT e ESC. Ogni volta che lo si fa, si passa al programma che segue secondo l'ordine del program selector fino a ritornare al programma di partenza e così via.

AGGIUNGERE E TOGLIERE PROGRAMMI DAL PROGRAM SELECTOR

Quando sul video è presente la maschera del program selector, sono attivi i tre tasti funzione F1, F5 e F10: il primo (Help) permette di accedere ad una funzione "guida" che spiega le funzionalità del programma e come utilizzare i tasti funzione; il

secondo (Refresh Switch List) aggiorna la colonna di destra nel caso in cui un programma terminato non sia ancora stato cancellato; il terzo (Update) permette di aggiungere, variare e cancellare programmi dalla colonna di sinistra. Se premiamo il tasto funzione F10, compare una nuova "cornice" (stentata traduzione dell'anglosassone "box", scatola) che contiene un "menu" di 4 voci strutturato come in figura 4.2.

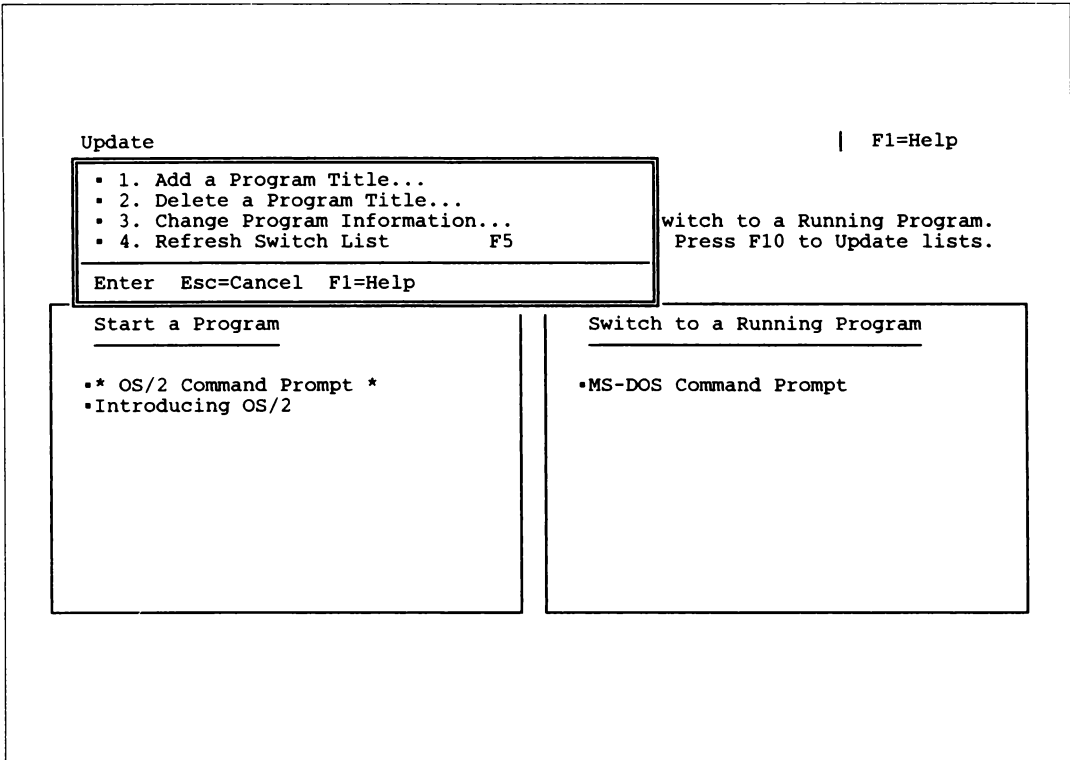



Figura 4.2

Per selezionare una voce, occorre premere ENTER per far apparire il fondo a colore invertito, dopodiché si opera come per le altre voci del program selector: una volta selezionata un'opzione, premendo ENTER la si avvia. La prima opzione permette di aggiungere un programma all'elenco della colonna Start a Program specificandone il nome, il pathname del programma da avviare quando questa voce viene selezionata e gli eventuali parametri richiesti dal programma. Questo grazie alla comparsa di un nuovo "box" dove è possibile impostare parametri mostrati in figura 4.3.

Figura 4.3

Alla fine del box, sono riassunte le funzioni di alcuni tasti: ENTER, ESC e il tasto funzione F1: quest'ultimo assolve la medesima funzione già documentata, mentre il tasto ESC permette di ritornare alla selezione precedente senza che niente di ciò che abbiamo eventualmente digitato venga "ricordato" dal computer. Il tasto ENTER, al contrario, esegue l'aggiornamento; quindi, per impostare i tre parametri, digitato il primo (cioè il nome che comparirà nell'elenco), passare al secondo usando i tasti direzionali e, una volta digitato anche questo parametro (cioè il pathname del programma), passare, se occorre, al terzo ancora con i tasti direzionali e digitarlo. In ogni caso, premendo ENTER, il box utilizzato scompare e il nuovo nome compare nella lista di sinistra da cui sarà possibile selezionarlo per attivare il programma di cui abbiamo digitato il nome.
La seconda opzione, tramite un diverso box, permette invece di togliere un nome, e conseguentemente la possibilità di attivare un programma, dalla lista di sinistra (figura 4.4).

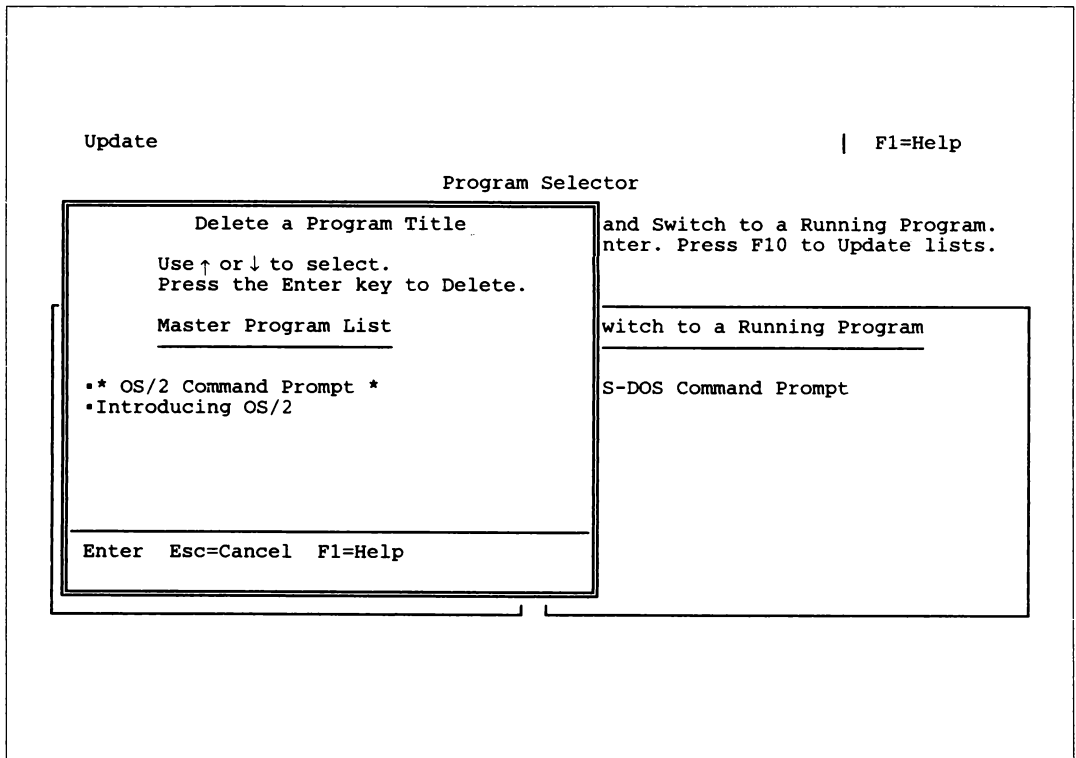



Figura 4.4

Anche qui, è documentata la funzione di ENTER, ESC e F1: da notare che il confermare con ENTER cancella la voce dall'elenco Start a Program, mentre premendo il tasto ESC si annulla la selezione, lasciando inalterato l'elenco suddetto.
La terza opzione, che presenta una maschera completamente analoga a quella della prima, permette di variare a piacere uno qualsiasi dei tre parametri già esposti: in questo modo, una voce nell'elenco, che fino a quel momento aveva mandato in esecuzione un programma, potrebbe avviarne uno diverso o, invece, un diverso nome potrebbe avviare il medesimo programma (figura 4.5).
Anche in questo caso, sono documentate le funzioni associate a ENTER, ESC e F1, che sono ovviamente analoghe a quelle della prima opzione, salvo il fatto che, premendo ESC, il nome e il programma impostati rimangono invariati nell'elenco.
L'ultima opzione è presente per comodità ma, come appare anche dalla maschera contenente le 4 funzioni, è equivalente alla pressione del tasto funzione F5.

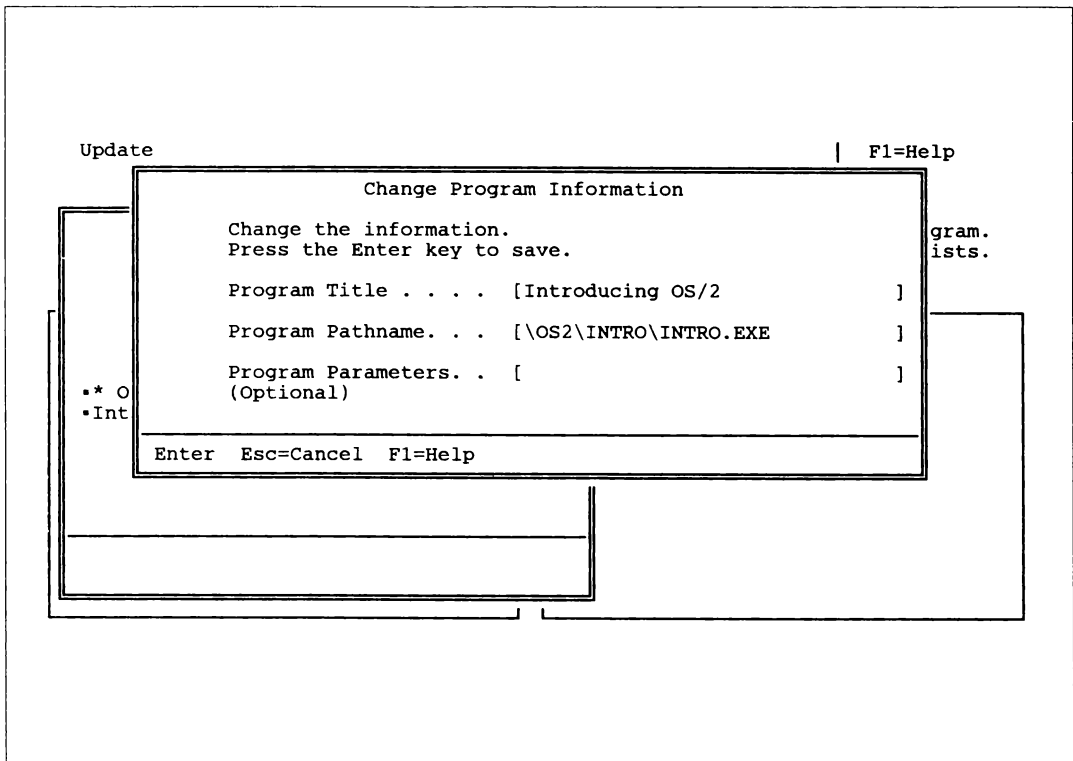



Figura 4.5

PROGRAMMI PER OS/2 E PROGRAMMI PER DOS

Una volta appresa la funzionalità della maschera del program selector e dei principi che governano il multitask in modo protetto e la sessione in modo reale, viene da chiedersi quali programmi possano essere inseriti nell'elenco di sinistra: in effetti, a tutt'oggi, i programmi che possono girare in modo protetto sono piuttosto scarsi. Per installare e utilizzare i programmi esistenti è pertanto necessario selezionare lo screen group del compatibility box ed eseguirle dal prompt che compare sullo schermo proprio come per un normale elaboratore che opera in DOS. Per sapere comunque se un determinato programma di cui siamo in possesso funzioni in modo protetto, la cosa migliore da farsi, che è fra l'altro anche la più semplice, è selezionare uno screen group "OS/2 Command Prompt" e da lì eseguire il programma suddetto: nel caso non abbia le caratteristiche necessarie, il sistema operativo, anziché avviare il programma, presenta questo messaggio:

"Messaggio OS/2": "pathname programma" cannot be run in *sigla costruttore* Operating System/2 mode

dove "Messaggio OS/2" è un identificatore composto da tre lettere più un numero di 4 cifre che identifica univocamente il tipo di messaggio, "pathname programma" è il nome del programma per cui volevamo verificare la compatibilità con il modo protetto e, infine, *sigla costruttore* individua il rilascio del prodotto OS/2 e può essere IBM o Microsoft.
Attualmente programmi che funzionino in modo protetto in OS/2 non se ne conoscono e anche il parco dei linguaggi di programmazione è piuttosto scarso (sono disponibili solo Assembler e C, e, al momento in cui questo libro va in stampa, un compilatore BASIC); è comunque possibile realizzare piccole procedure di utilità che vedremo nel capitolo 7 e che potranno essere incluse ed attivate dal program selector.

IMMISSIONE COMANDI: COME DIGITARE E CORREGGERE COMANDI

Una volta ottenuto il prompt, in qualsiasi screen group ci si trovi, occorre osservare alcune regole fondamentali per l'immissione dei comandi. Innanzitutto, occorre notare che l'estensione dei file che contengono programmi ricade obbligatoriamente in una delle seguenti classi:

COM	programmi eseguibili per i quali l'ambiente di utilizzo va verificato;
EXE	programmi eseguibili per i quali l'ambiente di utilizzo va verificato;
CMD	procedure realizzate dall'utilizzatore tramite l'accorpamento di programmi eseguibili ed eseguibile a sua volta solo in modo protetto;
BAT	procedure realizzate dall'utilizzatore tramite l'accorpamento di programmi eseguibili ed eseguibile a sua volta solo in modo reale.

Per far eseguire un comando a OS/2, è sufficiente digitarne il nome, esclusi il punto e l'estensione, fornire gli eventuali argomenti e parametri necessari e premere ENTER. Il primo controllo che viene effettuato dal sistema operativo riguarda l'esistenza di un comando con quel nome: se esiste, passa a verificare eventuali argomenti e opzioni, altrimenti manda un messaggio di errore (diverso in dipendenza dell'ambiente operativo) e ripresenta il prompt, pronto alla digitazione di un nuovo comando.
La ricerca del comando digitato, nel caso esistessero più comandi con il medesimo nome e diversa estensione, avviene con lo stesso ordine delle estensioni sopra esposte. Pertanto, se avessimo tre programmi con il medesimo nome, ed estensione rispettivamente .COM, .EXE e .BAT, digitando il nome comune ai tre programmi in emulazione DOS verrebbe eseguito quello con estensione .COM. Analogamente, se avessimo tre programmi con il medesimo nome ed estensione

rispettivamente .COM, .EXE e .CMD, digitando il nome suddetto in modo protetto, verrebbe eseguito quello con estensione .COM.
A questo proposito, è bene precisare che la ricerca del programma digitato avviene innazitutto secondo il path eventualmente specificato (vale a dire che noi possiamo digitare il nome del file contenente il nostro comando completo di drive e directory) tuttavia, nel caso noi avessimo fornito il solo nome e il comando non si trovasse nella directory corrente, prima di presentare un messaggio di errore il sistema operativo lo cercherebbe in tutte le directory indicate nella variabile di ambiente PATH (capitolo 5). Esiste in effetti un vero e proprio "ambiente operativo" associato ad ogni screen group e questo ambiente può contenere delle caratterizzazioni che valgono soltanto per quel particolare screen group e quindi per i programmi da esso attivati. Il PATH è un esempio di queste caratteristiche, dette "variabili di ambiente": il suo scopo è di fornire alcuni path per la ricerca comandi. Altra variabile di ambiente è PROMPT (capitolo 5), che permette di definire il formato appunto del prompt del sistema. Vedremo come definire variabili di ambiente nel prossimo capitolo, importante è tenere presente questa caratteristica che permette di adattare ogni screen group ad un particolare programma.
Una volta che il programma sia in esecuzione, esso verifica che eventuali argomenti e parametri da noi forniti e passatigli da OS/2 siano significativi per la/le funzioni che deve espletare: il proseguimento dell'elaborazione nel caso che uno o più parametri o opzioni non rientrino nei valori ammessi dipende interamente da chi ha realizzato il programma; solitamente esso si interrompe facendo apparire un messaggio di errore sul monitor.
Nella digitazione di un comando, ci sono alcuni tasti che hanno una funzione particolare:

←	questo tasto, detto "back space" (che significa "indietro di un carattere"), permette di ritornare indietro finché non si raggiunge la posizione dove si trova il carattere errato, dopodiché, corretto il carattere, occorre digitare nuovamente la parte successiva;
ENTER	questo tasto, chiamato anche RETURN, INVIO, CR e "↵ ", passa la linea da noi digitata al sistema operativo per l'esecuzione del comando richiesto;
ESC	Questo tasto interrompe la scrittura del comando e ci mette in condizione di riscriverlo completamente.

OS/2 ricorda tutto ciò che abbiamo digitato come ultimo comando e, così come ci permette di digitare nuovamente un diverso comando, ci offre la possibilità di utilizzare quanto precedentemente digitato grazie all'uso di tasti funzione:

F1	mostra un carattere ogni volta che viene premuto spostandosi avanti di un carattere nel comando memorizzato;

F2	seguito da un carattere, mostra tutti i caratteri del comando memorizzato fino al primo che incontra uguale a quello digitato;
F3	mostra tutti i caratteri dell'ultimo comando digitato, permettendo quindi di ripeterlo;
F4	seguito da un carattere, cancella tutti i caratteri del comando memorizzato fino al primo che incontra uguale a quello digitato;
F5	memorizza il comando che stiamo digitando senza eseguirlo, permettendoci così di effettuare le operazioni esposte come l'avessimo eseguito;
Del	cancella un carattere da quelli memorizzati;
Ins	i caratteri digitati di seguito vengono inseriti nel comando finché non viene premuto uno dei tasti funzione esposti.

Altri tasti, o meglio combinazioni di tasti, realizzate con il CTRL o l'ALT, sono riassunte nell'appendice A. È bene ricordare che i tasti hanno le funzioni elencate solo in presenza del prompt del sistema operativo: avviando un programma applicativo, questo prende possesso delle risorse del sistema, pertanto o disabilita completamente i tasti funzione o li dedica a funzioni diverse. Già lo stesso program selector, pur parte integrante del sistema operativo, assegna funzioni diverse ai tasti suddetti: anche queste sono comunque elencate nell'appendice A dove è specificato anche come attivarle, per i sistemi che ne sono dotati, tramite l'uso del mouse.

CAPITOLO CINQUE

I comandi di OS/2: comandi "esterni" ed "interni"

I comandi di OS/2 si dividono in "interni" ed "esterni": i primi costituiscono parte integrante del sistema operativo e pertanto la loro funzionalità è garantita dal semplice fatto che il computer su cui operiamo monta Operating System/2. I comandi esterni, invece, sono memorizzati sulla memoria di massa del calcolatore, ciascuno in un file il cui nome coincide con quello del comando che eseguono. È evidente che se nel disco del nostro Personal Computer non sono stati caricati tutti i comandi esterni, anche se stiamo operando in OS/2 non è detto che tutti i comandi funzionino.
La necessità di questa suddivisione si spiega con la semplice osservazione che se il sistema operativo contenesse tutti i programmi sarebbe enorme e la quantità di memoria RAM necessaria per utilizzarlo sarebbe improponibile; inoltre, poiché ciascun comando è interamente contenuto in un file che può essere singolarmente sostituito, in caso di malfunzionamento di un comando è sufficiente sostituire solo il file che lo contiene. Un'altra caratteristica da non trascurare in proposito è la possibilità che questa suddivisione offre di trasportare e distribuire il sistema operativo su più dischi: in effetti OS/2 viene distribuito su 4 dischi. L'unico scotto da pagare a questa scelta, è chiaramente il lavorio a cui l'elaboratore è sottoposto ogni volta che noi richiamiamo un comando esterno in quanto questo viene letto dal disco e caricato nella memoria RAM per poter essere eseguito, poi alla fine dell'esecuzione, la memoria RAM viene nuovamente liberata per il comando successivo e così di seguito.
Prima di passare all'esposizione analitica dei comandi, prendiamo spunto dalla precedente esposizione per notare che quello che chiamiamo programma, solitamente contenuto in un file memorizzato su disco, non è diverso né struttu-

ralmente (tipo di memorizzazione) né operativamente (metodo di avviamento del programma suddetto tramite il sistema operativo) dai comandi esterni: in effetti, un programma non è altro che un comando specializzato nella realizzazione di una particolare funzione (gestione testi, fogli elettronici, database, ecc.) che gode di un'unica fondamentale differenza dai comandi esterni di OS/2: occorre acquistarlo separatamente.
Il formato di tutti i comandi di OS/2 è regolato da norme precise: è obbligatorio il nome del comando, seguito da uno o più argomenti (la cui obbligatorietà è stabilita appunto dal comando), questi ultimi seguiti a loro volta da una o più opzioni.
I comandi che vedremo di seguito sono quelli che possono essere digitati in presenza del prompt di OS/2 e, per i comandi esterni, la dicitura "nome del comando", poiché identifica il nome del file che contiene il comando stesso, va interpretata come pathname del file suddetto, vale a dire che in risposta al prompt di OS/2 noi potremo digitare il nome del file che contiene il comando che vogliamo eseguire sia in forma di pathname relativo che assoluto (compresa eventualmente l'identificazione del drive), secondo quanto già esposto al capitolo 3. Nella esposizione analitica che segue, il nome del comando è privo del pathname, come cioè andrebbe digitato se il file che lo contiene si trovasse nella nostra directory corrente: questo permette un'uniformità nella notazione con i comandi interni ai quali non può essere premesso alcun pathname in quanto non risiedono in alcun file sul disco ma nella memoria RAM.
Anche gli eventuali argomenti del comando possono essere a loro volta dei nomi di file, pertanto anche per questi ultimi, occorre tenere conto delle medesime osservazioni fatte in merito al pathname dei comandi.
Le opzioni, infine, hanno normalmente tutte il medesimo formato che è caratterizzato da "/" seguita immediatamente da una lettera che rappresenta l'opzione stessa.
Per ciascun comando verrà specificato:

- l'ambiente di utilizzo (DOS, OS/2, entrambi);
- il tipo (esterno, interno);
- la corretta sintassi;
- almeno un esempio di utilizzo;
- riferimento a comandi correlati.

La struttura di esposizione per ogni comando è la seguente:

NOME
AMBIENTE
TIPO
DESCRIZIONE SINTETICA
SINTASSI
SPIEGAZIONE CON ESEMPI DI UTILIZZO

COMANDI CORRELATI

Il tipo di notazione usata per la sintassi di ciascun comando è la seguente:

- i caratteri che appaiono in maiuscolo corrispondono ai nomi dei comandi, o alle denominazioni delle opzioni e indicatori;
- i nomi scritti in corsivo identificano parametri al posto dei quali occorrerà inserire il valore adatto di volta in volta;
- i nomi degli argomenti opzionali sono racchiusi fra una coppia di [ ] che, nel caso di utilizzo dell'argomento non vanno digitate;
- i parametri che possono essere ripetuti sono seguiti sempre dalla notazione [...];
- diversi argomenti separati da | indicano la possibilità di scegliere uno solo di essi: se l'elenco è racchiuso fra [ ] la scelta è uno o nessuno, se è racchiuso fra { } la scelta di un elemento è richiesta.

ANSI

Ambiente: OS/2

Tipo: *Esterno*

Il comando ANSI permette, o impedisce, il controllo del video (colori, posizionamento cursore, ecc.) tramite le sequenze ANSI.

Sintassi ANSI [ON|OFF]

Il controllo del monitor del nostro calcolatore viene effettuato tramite sequenze di caratteri che traggono il loro nome dall'effetto che hanno quando vengono visualizzate: anziché mostrare un particolare carattere, provocano un particolare effetto sul video (cambiamento della posizione del cursore, inversione del colore di fondo, pulizia del video, emissione di un suono e molti altri). Questi effetti e le sequenze che li generano sono stati raccolti in uno standard (appendice B). Occorre osservare che è proprio il comando ANSI che stabilisce se queste sequenze otterranno sul monitor l'effetto dichiarato nello standard: in effetti, il comando ANSI senza alcun indicatore mostra il valore corrente; altrimenti lo altera in accordo con l'indicatore. Se non viene alterato, ANSI è ON quando il calcolatore viene acceso. Se, ad esempio, vogliamo utilizzare un programma per realizzazioni grafiche che richiedono che le sequenze ANSI non provochino sul video alcun effetto particolare, digiteremo ANSI OFF; e, per ripristinare il trattamento ANSI al termine del programma, digiteremo ANSI ON. Questo comando, utilizzabile solo in modo protetto, corrisponde all'assegnazione DEVICE=ANSI.SYS (capitolo 9) in modo reale.

Comandi correlati: DEVICE=ANSI.SYS in CONFIG.SYS, PROMPT

APPEND

Ambiente: DOS

Tipo: *Esterno*

Il comando APPEND consente di definire percorsi di ricerca (path), all'interno del file system, per file non contenenti programmi.

Sintassi APPEND [{;|path [;path][...]|/E}]

Con APPEND è possibile specificare una serie di una o più directory all'interno delle quali il sistema operativo ricerca i file che costituiscono, ad esempio, argomenti di altri comandi. Privo di argomenti, APPEND ci mostra l'elenco delle directory, o meglio dei path, precedentemente memorizzati separati da ";". Seguito dal solo ";" annulla qualsiasi path precedentemente memorizzato e infine, seguito da uno o più path, li memorizza, permettendone così l'uso al sistema operativo e ad altri comandi. Se, ad esempio, noi digitiamo APPEND immediatamente dopo l'accensione del calcolatore (attenzione, questo è un comando utilizzabile solo in modo DOS!) non vedremo apparire nulla sul monitor; se, successivamente digitiamo APPEND C:\ARCHIVI;C:\RISERVAT\ARCHIVI otterremo la memorizzazione di questi due path e, digitando nuovamente APPEND li vedremo comparire sul monitor. Se digitiamo APPEND ; ogni path verrà cancellato, cosa che possiamo verificare eseguendo nuovamente il comando. Particolare attenzione merita l'opzione /E: per utilizzarla, come si rileva dalla sintassi, occorre specificarla senza altri argomenti e, inoltre, occorre specificarla prima di ogni altra istruzione APPEND. L'utilizzo di questa funzione rende la memorizzazione successiva di path disponibile ai programmi applicativi in quanto i path così memorizzati entrano a far parte delle "variabili di ambiente" (capitolo 7). Per sapere se un particolare applicativo faccia uso di questa variabile di ambiente per determinare i path in cui reperire o memorizzare i file occorre consultare il manuale del programma stesso.

Comandi correlati: PATH, DPATH, SET

ASSIGN

Ambiente: DOS

Tipo: *Esterno*

Il comando ASSIGN permette di associare la lettera identificativa di un drive ad un'altro drive, dirottando così ogni operazione di lettura/scrittura su quest'ultimo.

Sintassi ASSIGN [drive da reindirizzare=drive di assegnazione][...]

Tramite il comando ASSIGN è possibile cambiare l'associazione lettera-drive, con l'evidente vantaggio di variare l'apparenza della configurazione del calcolatore ai programmi applicativi. Se, ad esempio, noi possedessimo un programma applicativo scritto per un elaboratore dotato dei soli drive a dischi A e B e, successivamente, dotassimo questo elaboratore di un hard disk C, tramite il comando:

```
ASSIGN A=C B=C
```

obbligheremmo il nostro applicativo a servirsi di quest'ultimo drive. Una volta effettuato un reindirizzamento, per annullarne l'effetto è sufficiente digitare ASSIGN senza alcun argomento. Occorre fare molta attenzione quando si usa il comando ASSIGN in quanto i comandi come BACKUP (vedi oltre) non sono più in grado di localizzare l'unità originale. I comandi FORMAT, DISKCOMP e DISKCOPY (vedi oltre) agiscono comunque sull'unità fisica indipendentemente dai reindirizzamenti effettuati.

Comandi correlati: SUBST, JOIN

ATTRIB

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando ATTRIB visualizza o modifica gli attributi associati ai file specificati.

Sintassi ATTRIB [{+R|-R}][{+A|-A}] pathname [/S]

Questo comando permette di gestire gli attributi "archivio" e "lettura" relativi a ciascun file memorizzato nella memoria di massa del nostro elaboratore. L'opzione +R attiva l'attributo di lettura, rendendo impossibile modifica e cancellazione del file corrispondente al pathname specificato, l'opzione -R la disattiva, rendendo possibili modifica e cancellazione. L'opzione +A attiva l'attributo archivio, utilizzato dai comandi BACKUP, RESTORE e XCOPY (vedi oltre) per effettuare copie selettive di file: in particolare questi comandi effettuano la copia dei file che hanno l'attributo archivio attivo e, al termine della copia, lo disattivano, analogamente all'opzione -A del comando ASSIGN. L'attributo archivio, oltre che dal comando che stiamo illustrando, viene attivato ogni qualvolta un file viene creato o modificato. Senza attributi, il comando ATTRIB mostra lo stato degli attributi del file specificato. Supponiamo di avere appena modificato il file "lettera.txt" che si trova nella directory "archivi" sul drive C (la nostra directory corrente): digitando ATTRIB lettera.txt sul monitor apparirà:

```
A C:\ARCHIVI\LETTERA.TXT
```

indicando che l'attributo archivio è attivo. Se volessimo impedire ogni altra operazione di modifica del contenuto di questo file, dovremmo digitare ATTRIB +R lettera.txt. Finché non rimuoviamo l'attributo lettura, il contenuto di lettera.txt non potrà in alcun modo essere variato e lettera.txt non potrà essere cancellato. L'opzione /S ripete il comando per tutti i file con il medesimo nome che si trovino in directory di livello inferiore rispetto a quella che contiene il file specificato. Se, nell'esempio precedente, supponiamo che esista una directory "mario" dislocata all'interno della directory archivi e che contenga a sua volta un file, "lettera.txt", digitando ATTRIB +R lettera.txt /S attiviamo l'opzione lettura su entrambi i file. Digitando ATTRIB lettera.txt sul monitor apparirà:

```
R A C:\ARCHIVI\LETTERA.TXT
```

Digitando ATTRIB mario\lettera.txt sul monitor apparirà:

R A C:\ARCHIVI\MARIO\LETTERA.TXT

Con questo comando possiamo realizzare una valida protezione dei nostri file da tentativi di modifica e cancellazione: poiché ATTRIB è un comando esterno, se noi attiviamo l'opzione lettura su tutti i file che non vogliamo siano modificati o cancellati e poi cancelliamo il file contenente il comando ATTRIB, impediamo a chiunque di disattivare l'opzione lettura e di conseguenza di modificare e cancellare i file suddetti. Questo metodo comunque non impedisce che i nostri file vengano ricopiati in quanto la copia non modifica il alcun modo l'originale: non ha quindi niente a che vedere con la sicurezza dei dati.

Attenzione: copiare il programma ATTRIB su un disco prima di cancellarlo altrimenti nemmeno noi saremo più in grado di disattivare l'opzione lettura!

Comandi correlati: BACKUP, RESTORE, XCOPY

BACKUP

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando BACKUP permette di effettuare una copia di uno o più file o di una intera directory, comprese quelle in essa contenute.

Sintassi BACKUP drive sorgente:[pathname] drive destinazione: [/M][/A] [/F][/S][/L:[pathname]] [/D:data] [/T:ora]

Con il comando BACKUP è possibile effettuare copie da un drive ad un altro di un numero di file anche eccedente la capacità del drive di destinazione poiché il comando, una volta riempito completamente un disco, ne richiede automaticamente un altro. Le copie effettuate con il comando BACKUP per trasferire file da un disco fisso a serie di floppy disk vengono dette copie di salvataggio perché servono solitamente per salvaguardare i file contenuti nel disco rigido in caso di danneggiamento di quest'ultimo: quando questo accade, non è più possibile reperire nessuno dei file memorizzati e quindi l'unica soluzione, dopo la riparazione del disco fisso, è quella di ripristinarne il contenuto tramite, appunto, le copie di salvataggio. Le differenti opzioni, che compaiono nella linea di comando precedente, hanno i seguenti significati:

/M Copia soltanto i file che abbiano l'attributo archivio attivo (vedi comando ATTRIB), disattivandolo non appena terminata la copia.

/A Normalmente, il comando BACKUP prescinde dall'eventuale contenuto del disco su cui effettua la copia; in altri termini l'esecuzione del comando di BACKUP cancella ogni precedente contenuto del disco di destinazione delle copie. Tramite questa opzione, invece, le copie effettuate dal comando BACKUP si accodano, sul disco di destinazione, a quelle già esistenti (naturalmente occorre che il disco di destinazione sia stato precedentemente usato con il comando BACKUP). Questa opzione non può comunque essere utilizzata per accodare file su dischi realizzati con il comando BACKUP di una versione di DOS inferiore a 3.3.

/F Formatta automaticamente il disco di destinazione (vedi comando FORMAT) Questa opzione non può essere utilizzata quando l'unità di destinazione è un disco rigido. Inoltre, occorre che la capacità del disco sia quella che il drive è in grado di formattare.

/S Se abbiamo specificato come pathname una directory che contiene a sua volta altre directory, con questa opzione il

comando BACKUP agisce anche su di esse, secondo quanto specificato dalle altre opzioni.

/L:pathname
: Quando è specificata questa opzione, la data, l'orario e l'elenco dei file che vengono copiati dal comando BACKUP vengono memorizzati nel file il cui nome è specificato dal parametro pathname e, in assenza di quest'ultimo, nel file BACKUP.LOG posto nella directory corrente. In ogni caso, se questo file esiste già, le informazioni vengono accodate a quelle precedentemente memorizzate. Per ogni file copiato, viene specificato un numero progressivo che rappresenta il disco di destinazione: questa numerazione è molto comoda nel caso in cui, effettuato un salvataggio che ha richiesto molti dischi, si debba ripristinare un solo file. Se nel comando di BACKUP era stata utilizzata l'opzione /L sappiamo su quale, fra i dischi utilizzati, si trova il file cercato.

/D:data
: Copia soltanto i file la cui data di creazione o di ultima modifica sia uguale o precedente a quella indicata dal parametro data. La data va specificata nel formato stabilito dal comando COUNTRY (nazione, capitolo 9) che, nel caso dell'Italia, è GG/MM/AA (cioè due cifre per il giorno, 2 per il mese e le ultime 2 dell'anno).

/T:ora
: Copia soltanto i file il cui orario di creazione o di ultima modifica sia uguale o precedente a quello indicata dal parametro ora. L'orario va sempre specificato nel formato OO:MM:SS (cioè due cifre per l'ora, 2 per i minuti primi e 2 per i minuti secondi).

I file di un salvataggio eseguito con il comando BACKUP devono, a priori, essere ripristinati per mezzo del comando RESTORE (vedi oltre). Il comando BACKUP è in grado di copiare indifferentemente da un disco rigido su di un floppy disk (e viceversa), da un disco rigido su di un altro disco rigido, e in generale fra tutti i tipi di memoria di massa che il calcolatore è in grado di gestire. Vediamo, ad esempio, il comando:

```
BACKUP C:\ A:/S/F/L:SALVA
```

Con questo comando tutti i file contenuti nel disco fisso C a partire dalla directory root per tutti i livelli del file system (opzione /S) vengono copiati su floppy disk. Ciascun floppy viene prima formattato (opzione /F) e, infine, nella directory root del drive C viene creato il file SALVA (o le informazioni gli vengono accodate se il file esisteva già) che conterrà data e orario di inizio del backup e la lista dei pathname dei file salvati, ciascuno seguito dal numero d'ordine del disco su cui viene copiato.

Comandi correlati: ATTRIB, RESTORE, XCOPY

BREAK

Ambiente: DOS

Tipo: *Interno*

Il comando BREAK definisce le modalità di arresto di un programma mediante l'uso dei tasti Ctrl e Break.

Sintassi BREAK [ON|OFF]

In emulazione DOS un programma può essere interrotto premendo simultaneamente i tasti Ctrl e Break (o Ctrl e C): l'interruzione del programma può avvenire immediatamente, oppure avvenire solo se il programma sta effettuando input (immissione dati da tastiera) o output (visualizzazione/stampa dati). In altri termini, questo comando permette (evita) l'interruzione successiva di un programma che non accetta dati da tastiera e non mostra alcunché su video e stampante, come ad esempio un compilatore. Digitando:

```
BREAK ON
```

comunichiamo al sistema operativo che intendiamo interrompere un programma in esecuzione, qualsiasi sia l'operazione in corso, nel momento in cui premiamo Ctrl e Break. Digitando:

```
BREAK OFF
```

comunichiamo al sistema operativo che desideriamo che l'esecuzione di un programma si arresti, quando noi premiamo Ctrl e Break, solo se detto programma è in attesa di dati da tastiera o sta visualizzando/stampando dati.
Il comportamento assunto dal sistema operativo rispetto ad un programma in esecuzione quando premiamo Ctrl e Break dipende dal valore inserito (o meno) nel file CONFIG.SYS (capitolo 9) per il comando corrispondente.
Alcuni programmi applicativi disabilitano la funzione di interruzione da parte dell'utente. Per questi programmi, il comando BREAK, comunque impostato, non produce alcun effetto.

Comandi correlati: BREAK in CONFIG.SYS

CHCP

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando CHCP visualizza, o cambia, il set di caratteri corrente.

Sintassi CHCP [codice]

Il parametro codice indica un valore numerico, corrispondente all'insieme di caratteri che si vuole rendere attivo. Questa scelta può essere effettuata fra due valori, ciascuno dei quali individua un diverso insieme di caratteri, purché questa possibilità sia stata prevista nel file CONFIG.SYS tramite il comando CODEPAGE (capitolo 9).
Si tratta, in definitiva, di poter scegliere come visualizzare particolari tipi di carattere utilizzati nelle varie nazioni, e di offrire la possibilità di effettuare questa scelta all'ambiente operativo: in pratica di poter attivare due differenti programmi ciascuno con il set di caratteri appropriato.
Digitando semplicemente CHCP, vengono visualizzati sia il valore del codice del set di caratteri corrente, sia quello dell'altro insieme. Se il comando CODEPAGE non è stato specificato nel file CONFIG.SYS, OS/2 presenta il seguente messaggio:

```
No code page support is available on the system
```

Il codice specificato dal comando CHCP deve essere necessariamente uno dei due codici definiti tramite il comando CODEPAGE. In caso contrario, il sistema operativo presenterà un messaggio di errore per il codice richiesto, presentando i due codici supportati.

Comandi correlati: CODEPAGE, COUNTRY e DEVINFO in CONFIG.SYS, SPOOL

CHDIR

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando CHDIR cambia la directory corrente per il drive specificato o visualizza il pathname assoluto della directory corrente.

Sintassi

```
CHDIR [drive:][path]
CD [drive:][path]
```

Questo comando, che può essere utilizzato indifferentemente nella forma estesa, CHDIR, che abbreviata, CD, permette di cambiare o visualizzare la directory corrente di ciascun drive di cui il sistema è dotato.
Supponendo di trovarci nella directory UTENTI, situata nella directory root sul drive C, digitando:

```
CD
```

il sistema operativo risponderà indicando il path completo della directory corrente, cioè:

```
C:\UTENTI
```

Per cambiare directory e spostarci, ad esempio, nella directory ARCHIVI, posta allo stesso livello nella stessa unità, digiteremo:

```
CD \ARCHIVI
```

oppure

```
CD ..\ARCHIVI
```

In entrambi i casi il sistema operativo non segnala nulla, si limita a ripresentare il prompt. Tuttavia, digitando nuovamente CD, mostrerebbe che la nostra directory corrente è divenuta C:\ARCHIVI.
Ora, supponendo di trovarci nella directory suddetta e di avere inserito nel drive A un disco nella cui directory root esiste una directory DATI, digitando:

```
CD A:\DATI
```

quello che otteniamo è che la directory DATI divenga la directory corrente per il drive A, ma la nostra directory corrente rimane comunque C:\ARCHIVI. Digitando

però:

CD A:

il sistema operativo risponderà indicando il path completo della directory corrente se l'utente si trovasse sul drive A; e cioè:

A:\DATI

In altri termini, esiste una directory corrente per ciascun drive collegato al sistema; cambiando drive, ci troveremo nella directory corrente per quel drive. Se, nella situazione precedentemente descritta, digitiamo:

A:

l'effetto è quello di passare sul drive A; infatti digitando CD otterremmo dal sistema operativo la risposta A:\DATI, mentre digitando CD C: otterremmo dal sistema operativo la conferma che la directory corrente del drive C: è C:\ARCHIVI. In definitiva, possiamo parlare di directory corrente per l'utilizzatore, in quanto dipende dal drive selezionato da quest'ultimo, e di directory corrente per ciascun drive, in quando selezionata con il comando CD (o assunta per default uguale alla directory root). La directory corrente dell'utilizzatore coincide ovviamente con la directory corrente del drive selezionato dall'utilizzatore. Notiamo che il prompt del sistema operativo indica qual è il drive corrente.

Comandi correlati: MKDIR, RMDIR, TREE

CHKDSK

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando CHKDSK effettua una verifica sullo stato del disco e sulla sua integrità.

Sintassi CHKDSK [drive:][pathname] [/F] [/V]

Con questo comando è possibile verificare l'integrità e soprattutto la reperibilità dei dati memorizzati nella memoria di massa del nostro elaboratore. Come abbiamo già visto (capitolo 3), un certo insieme di dati (archivio), di istruzioni (programma), e in generale tutto ciò che viene memorizzato sui dischi va sotto il nome generico di file: questo nome è l'unico legame che abbiamo con il contenuto del file stesso. Il sistema operativo mantiene una sorta di indice (vedi comando DIR, più oltre) di questi nomi, a ciascuno dei quali corrisponde una "mappa" per la localizzazione del contenuto del file: un'alterazione di questa mappa, anch'essa memorizzata sul disco, renderebbe evidentemente inaccessibile il contenuto di uno o più file. Il sistema operativo mantiene anche una mappa dello spazio disponibile sul disco: è evidente che esiste una stretta relazione fra l'indice dei file e lo spazio disponibile sul disco: la "somma" delle due mappe deve corrispondere all'intero disco: questo è quanto il comando CHKDSK è in grado di verificare ed, eventualmente, ripristinare.
In mancanza dell'indicazione drive CHKDSK verifica il drive corrente, altrimenti quello specificato. Se forniamo il pathname di un file, CHKDSK ci dice se detto file è memorizzato in maniera contigua sul disco o meno. Sono inoltre previste le seguenti opzioni:

/F In caso di incongruenza fra l'indice dei file e la mappa dello spazio disponibile, la presenza di questa opzione abilita la correzione del problema riscontrato. Il caso più frequente è quello di un file che non compare nell'indice, rilevato perché nella mappa dello spazio disponibile risulta occupata una certa porzione di disco che però non appartiene a nessuno dei file il cui nome e relativa mappa del contenuto costituiscono l'indice. La presenza di questa opzione fa si che venga creato nell'indice un nome la cui mappa sia quella del file "orfano".

/V Causa la visualizzazione del nome di tutti i file presenti sul disco, al momento della verifica.

Il comando CHKDSK, indipendentemente dalle opzioni usate (che possono eventualmente aggiungere altri messaggi) presenta comunque sempre una esposizione

dell'utilizzo del disco:

```
Volume MS-OS2 created 2 Feb 1988 14:28~22142976 bytes total disk space

270098 bytes in 3 hidden files
172032 bytes in 83 directories
16412672 bytes in 749 user files
5288174 bytes available un disk
```

(Il nome del "volume", la data della "creazione" e le cifre sono casuali). Se questo comando viene eseguito in emulazione DOS, aggiunge in coda alla tabella precedente un dato riguardante la quantità di memoria RAM disponibile:

```
[DOS mode storage report]
    655328 bytes total storage
    530816 bytes free
```

Una nota importante riguardo a questo comando riguarda le modalità della sua esecuzione: in particolare, dobbiamo evitare di eseguirlo quando vi sono altri programmi attivi, magari in altri screen group (capitolo 4), in quanto un programma che accede a un archivio non aggiorna la mappa dello spazio usato per quel file nell'indice se non al termine del programma stesso, pertanto CHKDSK rileverebbe delle incongruenze e, al limite, se eseguito con l'opzione /F, altererebbe in modo irrimediabile l'archivio in uso.

Comandi correlati: RECOVER

CLS

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando CLS pulisce lo schermo.

Sintassi CLS

Dopo la pulizia dello schermo, il prompt si trova nell'angolo superiore sinistro del monitor.

CMD

Ambiente: OS/2

Tipo: *Esterno*

CMD è il nome del programma che presiede al dialogo con l'utilizzatore nelle sessioni avviate in modo protetto.

Sintassi CMD [pathname] [/Ccomando|/Kcomando]

Il comando CMD viene attivato dal program selector (capitolo 4) ogni volta che noi avviamo un nuovo screen group in modo protetto. Questo comando, detto "command processor", è responsabile del prompt e del trasferimento dei comandi da noi digitati, dopo averne verificata la congruenza, al sistema operativo perché li esegua.
Il command processor è composto da un modulo "residente", cioè trasferito dal disco nella memoria RAM in modo permanente, e da un modulo "transitorio", cioè richiamato dal disco solo quando è necessario: pathname serve ad indicare al sistema operativo dove si trova il programma che contiene la parte transitoria quando questa non risiede nella directory root.
Niente vieta che il comando che noi chiediamo di eseguire sia proprio CMD: l'utilità di questa richiesta sta nel fatto che questo nuovo command processor, detto "figlio", possiede le stesse variabili di ambiente del cosiddetto "padre", tuttavia ogni cambiamento effettuato su queste variabili non ha alcun effetto sul command processor originale una volta terminato il nuovo CMD.
È possibile specificare due opzioni:

/C comando Il nuovo command processor viene attivato, a sua volta esegue comando (che può essere a sua volta dotato di argomenti e opzioni) e, terminato detto comando, termina anch'esso, restituendo il controllo al CMD originale.

/K comando Come per il caso precedente, con l'unica differenza che terminato il comando, rimane attivo il command processor figlio, che restituirà il controllo al padre solo quando digiteremo il comando EXIT (vedi oltre).

Il program selector, ogni volta che noi attiviamo una sessione in modo protetto, attiva il comando seguente:

```
CMD /K OS2INIT.CMD
```

cioè avvia il command processor con l'opzione /K affinché rimanga attivo finché

non digitiamo EXIT e facendogli inoltre eseguire il comando OS2INIT.CMD (capitolo 7).

Comandi correlati: COMMAND, EXIT, START

COMMAND

Ambiente: DOS

Tipo: *Esterno*

COMMAND è il nome del programma che presiede al dialogo con l'utilizzatore nella sessione che emula il DOS.

Sintassi COMMAND [pathname] [/Ccomando|/P] [/E:dimensione]

Il comando COMMAND viene attivato quando, dal program selector, selezioniamo lo screen group in emulazione DOS. Questo comando, detto "command processor", è responsabile del prompt e del trasferimento dei comandi da noi digitati, dopo averne verificata la congruenza, al sistema operativo perché li esegua.
Come per il command processor del modo protetto, anche questo command processor è composto da un modulo "residente", cioè trasferito dal disco nella memoria RAM in modo permanente, e da un modulo "transitorio", cioè richiamato dal disco solo quando è necessario: pathname serve a indicare al sistema operativo dove si trova il programma che contiene la parte transitoria quando questa non risiede nella directory root.
Anche in emulazione DOS, noi possiamo eseguire il comando COMMAND: l'utilità di questa richiesta sta nel fatto che questo nuovo command processor, detto "figlio", possiede le stesse variabile di ambiente del cosiddetto "padre", tuttavia ogni cambiamento effettuato su queste variabili non ha alcun effetto sul command processor originale, una volta terminato il nuovo COMMAND.
È possibile specificare tre opzioni:

/C comando — Il nuovo command processor viene attivato, a sua volta esegue comando (che può essere a sua volta dotato di argomenti e opzioni) e, terminato detto comando, termina anch'esso, restituendo il controllo al COMMAND originale.

/P — Questa opzione specifica che il nuovo command processor deve rimanere attivo anche dopo l'esecuzione di un eventuale comando: il controllo ritornerà al command processor originale solo quando digiteremo EXIT.

/E:dimensione — Questa opzione permette di definire la dimensione in byte dello spazio riservato alle variabili di ambiente dal nuovo command processor. Può essere specificato qualsiasi valore da 160 a 32768: viene arrotondato per eccesso al successivo multiplo di 16; il valore di default è 160 byte.

Quando selezioniamo l'emulazione DOS dal program selector, viene eseguito il

comando:

```
COMMAND /P
```

cioè viene avviato il command processor con l'opzione /P affinché rimanga attivo. È da notare che il command processor di emulazione DOS, una volta avviato, non può più essere cancellato: l'unico modo per toglierlo di mezzo è far ripartire il calcolatore.

Comandi correlati: CMD, EXIT

COMP

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando COMP confronta il contenuto di due o più file segnalandone le differenze.

Sintassi COMP [pathname1] [pathname2]

Il comando COMP permette di confrontare il contenuto del file specificato da pathname1 con il contenuto del file specificato da pathname2. Se il comando viene attivato senza alcun nome di file, la richiesta dei nomi viene effettuata dal comando stesso. Utilizzando i caratteri "?" e "*" nella specifica dei pathname, è possibile eseguire la comparazione di più file.
Tutte le combinazioni di specifica dei pathname sono possibili: possiamo ad esempio digitare:

```
COMP *.TXT B:
```

il risultato sarà di confrontare tutti i file che hanno estensione .TXT nella directory corrente con i file che hanno il medesimo nome posti nel disco A. Supponendo che nella directory root del nostro disco fisso esistano due directory chiamate ARCHIVI e DATI, digitando:

```
COMP \ARCHIVI \DATI
```

otterremo la comparazione di tutti i file della prima directory con quelli della seconda.
Dalla comparazione più semplice alla più complessa, al termine il comando COMP chiede se vogliamo effettuare altri confronti, richiedendo, in caso di risposta affermativa, i nomi dei file da confrontare e ripetendo il ciclo.
Il confronto fra i file viene effettuato byte per byte. I nomi dei file confrontati vengono visualizzati man mano che si svolge la procedura.
È possibile paragonare tra loro anche file di diverse dimensioni. In questo caso, il sistema operativo presenta il seguente messaggio:

```
Files are different sizes, do you wish to continue (Y/N)?
```

In caso di risposta affermativa (Y), il confronto viene effettuato per la lunghezza del file più corto.
Quando due file risultano essere essere assolutamente identici, questo viene segnalato tramite il messaggio:

Files compare OK

Quando, invece, i due file sono differenti, viene segnalata la posizione a partire dall'inizio dei due file in cui il contenuto è diverso, e i rispettivi valori in quella posizione:

Compare error at OFFSET posizione
pathname1 = byte
pathname2 = byte

dove posizione, espresso in forma esadecimale (la numerazione esadecimale è una numerazione in base 16, dove cioè sono necessarie 16 unità prima di arrivare alla "decina" seguente), corrisponde alla posizione occupata da byte (anche questo espresso in forma esadecimale), calcolata a partire dall'inizio del file. Alla decima differenza, il confronto si interrompe e appare il messaggio:

10 Mismatches - ending compare

e il comando COMP prosegue la comparazione dei due file successivi o richiede se si vuole effettuale la comparazione di altri file.

Comandi correlati: DISKCOMP

COPY

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando COPY consente di copiare e compattare file.

Sintassi COPY pathname [/A|/B] [+pathname [/A|/B]][...]
[pathname [/A|/B][/V]]

Il comando COPY permette moltissime combinazioni: esaminiamo prima le opzioni e poi esaminiamo i diversi casi possibili. Il significato delle opzioni è il seguente:

/A Il file viene considerato come file di testo: se l'opzione riguarda un file sorgente, questo significa che la copia continua finché non viene incontrato un carattere di fine file; se riguarda il file destinazione, viene aggiunto un carattere di fine file al termine della copia. Il "fine file" è un byte che ha un valore particolare, detto "control Z" in quanto è lo stesso carattere che si ottiene premendo contemporaneamente i tasti Ctrl e Z.

/B Questa opzione ha praticamente l'effetto opposto alla precedente: se riguarda un file sorgente, la copia continua fino alla fine del file, indipendentemente dai caratteri in esso contenuti; se riguarda il file destinazione, quest'opzione previene l'aggiunta di un carattere di fine file al termine della copia.

/V Verifica la copia effettuata rileggendola a mano a mano che viene eseguita. Questa opzione provoca un rallentamento dell'operazione di copia ed è consigliabile utilizzarla solo in caso di dubbi sulla qualità del disco impiegato come destinazione.

Le opzioni /A e /B valgono per il pathname che le precede e rimangono valide per tutti i pathname seguenti fino alla opzione successiva o, meglio, al pathname che precede l'opzione successiva.
Esaminiamo ora una ad una le possibili combinazioni del comando COPY:

- Specificando un pathname completo di drive, path e nome del file, sia per la destinazione che per la sorgente, il risultato è di avere una copia del file sorgente con un altro nome in un'altra directory di un altro drive. Se, ad esempio, vogliamo copiare il file VENDITE.GEN situato nella directory

VENDITE, posta sul disco nel drive B nella directory STORICO posta sul disco nel drive A con il nome STORICO.GEN, dobbiamo digitare:

COPY B:\VENDITE\VENDITE.GEN A:\STORICO\STORICO.GEN

Chiaramente, se la directory VENDITE fosse stata la nostra directory corrente e il drive B il nostro drive corrente, il comando si sarebbe ridotto a:

COPY VENDITE.GEN A:\STORICO\STORICO.GEN

Viceversa, se fosse stato A il nostro drive corrente e STORICO la nostra directory corrente, il comando sarebbe stato:

COPY B:\VENDITE\VENDITE.GEN STORICO.GEN

Infine, supponendo che il nostro drive corrente sia A, ma che la nostra directory non sia STORICO, avremmo digitato:

COPY B:\VENDITE\VENDITE.GEN \STORICO\STORICO.GEN

In tutti i casi, nella directory STORICO del drive B si troverà un file STORICO.GEN con il medesimo contenuto del file VENDITE.GEN posto nella directory VENDITE del drive A.

- Specificando il pathname assoluto per la sorgente, ma solo un path composto di disco e/o directory per la destinazione si ottiene una copia del file originale in un altro disco e/o path ma con il medesimo nome. Se, ad esempio, vogliamo copiare il file VENDITE.GEN situato nella directory VENDITE, posta sul disco nel drive B, nella directory STORICO, posta sul disco nel drive A, dobbiamo digitare:"COPY B:\VENDITE\VENDITE.GEN A:\STORICO". Chiaramente, se la directory VENDITE fosse stata la nostra directory corrente e il drive B il nostro drive corrente, il comando si sarebbe ridotto a:

  COPY VENDITE.GEN A:\STORICO

  Viceversa, se fosse stato A il nostro drive corrente e STORICO la nostra directory corrente, poiché non abbiamo la necessità di specificare il nome del file destinazione, in quanto uguale a quello sorgente, e il path è quello corrente, il comando sarebbe stato:

  COPY B:\VENDITE\VENDITE.GEN

questo è cioè un caso in cui per effettuare la copia è sufficiente specificare un solo pathname: quello del file sorgente. Infine, supponendo che il nostro drive corrente sia A, ma che la nostra directory non sia STORICO, avremmo digitato:

```
COPY B:\VENDITE\VENDITE.GEN \STORICO
```

In tutti i casi, nella directory STORICO del drive B si troverà un file VENDITE.GEN perfettamente identico a quello posto nella directory VENDITE del drive A. In questo caso, l'utilizzo dei caratteri "?" e "*" è particolarmente utile per copiare una serie di file in un altro disco/directory. Se, ad esempio, supponiamo di voler copiare tutti i nostri file di testo (che abbiamo creato con estensione TXT) posti nella directory root del disco C, che è anche la nostra directory corrente, nella directory TESTI posta a sua volta nella directory APPUNTI del medesimo disco, digiteremo:

```
COPY *.TXT APPUNTI\TESTI
```

Il risultato di questo comando è la duplicazione di tutti i file, posti nella nostra directory corrente, nella directory TESTI, posta a sua volta nella directory APPUNTI la quale, a sua volta, si trova nella nostra directory corrente.

- Specificando, nel pathname di destinazione, solo il drive o il drive con il nome del file, si ottiene la copia del file sorgente, con il medesimo nome o con quello digitato, nella directory corrente per il drive di destinazione. Se, ad esempio, STORICO è la directory corrente del drive B, digitando:

```
COPY A:\VENDITE\VENDITE.GEN B:STORICO.GEN
```

si ottiene lo stesso risultato proposto nella prima casistica, mentre digitando:

```
COPY A:\VENDITE\VENDITE.GEN B:
```

si ottiene lo stesso risultato proposto nella seconda casistica.
Anche in questo caso è naturalmente possibile usare i caratteri "?" e "*": se vogliamo ripetere un esempio analogo all'ultimo, copiando i nostri file di testo nella directory APPUNTI che è la directory corrente del drive A, digiteremo:

```
COPY *.TXT A:
```

e il risultato sarà proprio quello di copiare tutti i file con estensione .TXT

presenti nella nostra directory corrente nella directory APPUNTI del drive A.

- Specificando una directory come pathname sorgente, si ottiene la copia di tutti i file contenuti in quella directory (ma non quelli contenuti in alcuna directory che si trovi al suo interno) nella directory di destinazione. Se viene specificato un pathname di destinazione, è necessario che individui una directory a sua volta, in quanto, se invece individuasse un file, si otterrebbe la copia di tutti i file della directory sorgente nell'unico file di destinazione, e quindi la copia sarebbe inutilizzabile. Supponendo di voler copiare tutti i file della directory VENDITE degli esempi citati nella directory STORICO, dovremmo digitare:

  ```
  COPY A:\VENDITE B:\STORICO
  ```

  Questo comando equivale a COPY A:\VENDITE*.* B:\STORICO, il cui principio è esposto durante la seconda casistica, in questa forma tuttavia è più compatto, soprattutto se adottiamo qualche default su drive e directory.

- Specificando diversi pathname separati dal segno "+" come sorgenti, si ottiene come copia di destinazione un file che è, potremmo dire, la "somma" dei file sorgente, cioè è composto dai contenuti dei file sorgente posti uno di seguito all'altro nell'ordine in cui sono specificati. Se il nome del file di destinazione compare anche fra quelli sorgente, quest'ultimo non viene preso in considerazione e il sistema operativo visualizza un messaggio di errore. In altri termini, il sistema operativo confronta ogni nome della sequenza di file sorgente con il nome del file destinazione: se i nomi sono diversi, il file sorgente viene "accodato" al file destinazione; se sono uguali, il file sorgente viene saltato e il sistema operativo passa al successivo file della sequenza.

Per fare un esempio, supponiamo di avere una serie di tabelle, ciascuna delle quali contenga il resoconto delle vendite di un mese, ciascuna con il nome del mese ed estensione TAB. Se alla fine dell'anno volessimo raccogliere il resoconto di tutte le vendite del 1987 in un unico file, che chiameremmo 1987.TAB, digiteremo:

```
COPY GENNAIO.TAB + FEBBRAIO.TAB + MARZO.TAB + APRILE.TAB +
MAGGIO.TAB + GIUGNO.TAB + LUGLIO.TAB + AGOSTO.TAB + SETTEMBRE.TAB
+ OTTOBRE.TAB + NOVEMBRE.TAB + DICEMBRE.TAB 1987.TAB
```

Il risultato di questo comando è il file 1987.TAB che contiene le 12 tabelle nell'ordine assegnato. Notiamo che se manca l'indicazione del file di destinazione (cioè non c'è alcun file che non sia separato dall'altro dal segno +), viene preso il nome del primo file che compare nella lista: in questo caso il comando COPY

espleta una vera e propria funzione di accodamento. Una variazione dell'esempio precedente può spiegare questa funzione: supponiamo di voler raccogliere i dati trimestralmente sotto il nome del primo mese del trimestre. Il file GENNAIO.TAB esiste già e contiene i dati di gennaio: per "accodargli" i dati di febbraio e marzo digiteremo:

```
COPY GENNAIO.TAB + FEBBRAIO.TAB + MARZO.TAB
```

Poiché non abbiamo specificato alcun file di destinazione, FEBBRAIO.TAB e MARZO.TAB vengono accodati a GENNAIO.TAB. Per concludere la trattazione del comando COPY, osserviamo che l'opzione /A è assunta di default nel caso di accodamenti, mentre l'opzione /B è assunta di default negli altri casi.

Comandi correlati: DISKCOPY, RENAME, VERIFY, XCOPY

CREATEDD

Ambiente: OS/2

Tipo: *Esterno*

CREATEDD è un comando di manutenzione per creare un dischetto su cui successivamente effettuare un "dump" della memoria.

Sintassi CREATEDD drive:

Per scopi di verifica del funzionamento di OS/2 rispetto a particolari sollecitazioni, è possibile registrare su dischi il contenuto della memoria RAM ad un determinato punto dell'elaborazione (il termine anglosassone "dump" è stato coniato per indicare la "fotografia" della memoria ad un dato istante). Per far ciò, è necessario aver predisposto un apposito disco con questo comando. La quantità di memoria occupata, e quindi di cui eseguire il "dump", può richiedere più di un solo disco, tuttavia CREATEDD va fatto solo per il primo disco della serie; i successivi, che il sistema richiederà se il primo risultasse insufficiente, è vanno formattati normalmente (vedi comando FORMAT, più oltre). La sintassi del comando, peraltro semplice, richiede il nome del drive in cui è stato inserito il disco da cui iniziare il dump.
Questo comando si limita a preparare un disco per la funzione di dump: la funzione vera e propria può essere attivata in qualsiasi momento premendo contemporaneamente i tasti Ctrl, Alt e, tenendoli premuti, premendo due volte di seguito il tasto NumLock. Notiamo che il dump della memoria può essere effettuato indifferentemente in DOS e in OS/2 e, inoltre, che questa operazione arresta tutte le attività del calcolatore che, al termine del dump, andrà spento e riacceso o, comunque, "reinizializzato". È buona norma pertanto, se si rende necessario effettuare un dump della memoria (potrebbe, ad esempio, sorgere la necessità di mandare il dump della memoria alla casa costruttrice per risolvere problemi di malfunzionamento di OS/2 con un particolare applicativo), farlo quando il calcolatore non ha altre attività incorso.
Il dump della memoria avviene con la richiesta, da parte del sistema operativo, del disco creato con questo comando, dopodiché vengono richiesti altri dischi finché tutta la memoria non è stata scaricata e, infine, viene chiesto nuovamente il disco iniziale con il quale si chiude il dump e il calcolatore si arresta.

Comandi correlati: TRACE e TRACEBUF in CONFIG.SYS, TRACE, TRACEFMT

DATE

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando DATE permette di visualizzare e di modificare la data corrente.

Sintassi DATE [data]

Il comando DATE, digitato senza alcun argomento, mostra la data nel formato specificato per il paese selezionato dal comando COUNTRY in CONFIG.SYS (capitolo 9) e si predispone ad accettarne la variazione nel medesimo formato. Indipendentemente dal formato scelto, che riguarda la sequenza giorno, mese e anno e il carattere di separazione per queste tre cifre, le regole per l'immissione della data sono le seguenti:

- il giorno va indicato con un numero compreso fra 1 e 31 con l'avvertenza che il sistema operativo verifica, rispetto a mese ed anno, la congruenza del valore immesso;
- il mese va indicato con un numero compreso fra 1 e 12;
- l'anno può essere indicato con un numero di 4 cifre compreso fra 1980 e 2079, oppure con un numero di due cifre compreso fra 00 e 99, con l'avvertenza che l'intervallo che va da 00 a 79 sottintende gli anni da 2000 a 2079, mentre l'intervallo che va da 80 a 99 sottintende gli anni da 1980 a 1999.

Se il 4 febbraio 1988, con il formato italiano selezionato, digitiamo DATE compare il seguente messaggio:

```
The current date is: Thu 4/2/88
Enter the new date (dd-mm-yy):
```

cui possiamo rispondere digitando una nuova data o premendo semplicemente ENTER per confermare quella inserita. Il comando DATE può anche essere seguito direttamente dalla nuova data da impostare: in questo caso non comparirà alcuna richiesta, salvo che non si commetta un errore nell'impostarla.

Comandi correlati: COUNTRY in CONFIG.SYS, TIME

DEL
ERASE

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando DEL o ERASE cancella uno o più file.

Sintassi
```
DEL pathname [...]
ERASE pathname [...]
```

Pathname può contenere i caratteri "?" e "*". La specifica di più pathname di file da cancellare è riservata al modo protetto. Le diciture DEL e ERASE sono perfettamente equivalenti.

Il comando:

```
DEL *.*
```

permette di cancellare tutti i file della directory corrente. Prima che inizi la cancellazione, viene visualizzata una richiesta di conferma:

```
ARE YOU SURE?
```

Rispondendo Y la cancellazione viene effettuata, altrimenti no. Questo messaggio viene visualizzato anche se il pathname dato al comando DEL è quello di una directory, in quanto una risposta positiva cancella l'intero contenuto della directory stessa.
Il comando DEL non cancella in alcun caso directory, siano esse "vuote" o meno. Occorre notare che la limitazione del comando DEL in ambiente DOS non è di cancellare un solo file per volta ma di accettare un solo pathname: questo, tuttavia, potendo contenere i caratteri "?" e "*", può causare la cancellazione di molti file.

DETACH

Ambiente: OS/2

Tipo: *Interno*

Sintassi DETACH pathname [argomento][...]

Il pathname richiesto è quello di un programma da eseguire in "background". Background, letteralmente "sottoterra", è un termine comune per indicare il fatto che un programma avviato in questo modo non è legato ad alcuna maschera video e la sua esecuzione continua indipendentemente da altri fattori esterni: terminerà soltanto quando finirà le elaborazioni che gli sono state affidate o al momento dello spegnimento del calcolatore.
Gli argomenti, quando specificati, devono essere significativi per il programma, in quanto è da questo che vengono presi in considerazione, non da DETACH.
È possibile eseguire in background anche programmi interattivi, cioè che richiedono dati da tastiera e li mostrano sul video: poiché ciò non è possibile per i programmi attivati con DETACH, si usa la tecnica del "reindirizzamento" dell'input e dell'output (capitolo 7).

Comandi correlati: RUN in CONFIG.SYS

DIR

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando DIR visualizza i nomi dei file in una directory.

Sintassi DIR [drive:][pathname] [...] [/P] [/W]

Il comando DIR permette di visualizzare i nomi dei file ed altre informazioni: lunghezza totale in bytes, data e ora di creazione o ultima modifica.
Digitando DIR senza alcun argomento, si ottiene l'elenco dei file della nostra directory corrente. Le opzioni hanno il significato seguente:

/P Questa opzione è utile quando i file di cui elencare i nomi occupano più di una schermata: con questa opzione, l'elencazione si arresta dopo aver riempito lo schermo, presentando nell'ultima linea il messaggio:

```
Press any key when ready.
[premere qualsiasi tasto per proseguire]
```

premendo un tasto qualsiasi l'elencazione continua fino a riempire nuovamente lo schermo, dopodiché compare nuovamente il suddetto messaggio e così di seguito fino al termine dell'elenco.

/W Questa opzione presenta solo i nomi dei file, senza altre informazioni, posti su cinque colonne: permette, in altri termini, di vedere i nomi di un gran numero di file nella stessa maschera video.

Il comando DIR mostra sempre, indipendentemente dai pathname e dalle opzioni impostate, la lettera identificativa del drive, il nome del disco, il nome della directory di cui elenca i file e, al termine dell'elenco, il numero totale di file elencati e lo spazio libero rimanente su quel disco. Se, ad esempio, trovandoci nella directory root del disco chiamato "OS2" nel drive C, digitiamo:

```
DIR
```

otteniamo:

```
Volume in drive C in OS2
Directory of C:\

OSO001     MSG      65077   15/12/87   0:00
OSO001H    MSG      84201   15/12/87   0:00
CONFIG     SYS         73   15/12/87   0:00
COUNTRY    SYS      14644   15/12/87   0:00
CLOCK01    SYS       2762   15/12/87   0:00
CLOCK02    SYS       3188   15/12/87   0:00
DISK01     SYS      20197   15/12/87   0:00
DISK02     SYS      20329   15/12/87   0:00
KBD01      SYS      16945   15/12/87   0:00
KBD02      SYS      15650   15/12/87   0:00
PRINT01    SYS       7683   15/12/87   0:00
PRINT02    SYS       9018   15/12/87   0:00
SCREEN01   SYS       1583   15/12/87   0:00
SCREEN02   SYS       1988   15/12/87   0:00
HARDERR    EXE      16304   15/12/87   0:00
SWAPPER    EXE       4150   15/12/87   0:00
OS2           <DIR>          2/02/88  11:15

     17 File(s)  18245604 bytes free
```

Innanzitutto notiamo che i file veri e propri sono caratterizzati da nome ed estensione (che nell'esposizione del comando DIR appaiono separati da spazi ma che occorre sempre digitare separati dal "."), oltre che da lunghezza, data e orario di creazione o ultima modifica, mentre le directory sono caratterizzate dalla sigla DIR racchiusa fra < e > e, anche per esse, sono indicate data ed orario di creazione o ultima modifica, tenendo presente che, per una directory, la modifica consiste nell'aggiunta o nella cancellazione di un file in essa contenuto.
Per il solo modo protetto, è possibile fornire diversi pathanme: in questo modo, se i pathname forniti individuano drive diversi, otterremo più elenchi, ciascuno nel formato suddetto. I pathname forniti a DIR possono contenere i caratteri "?" e "*" per permettere l'individuazione di gruppi omogenei di file. Digitando ad esempio:

```
DIR \OS2\*.DCP
```

otteniamo:

```
Volume in drive C in OS2
Directory of C:\OS2

KEYBOARD   DCP      85917   15/12/87   0:00
VIOTBL     DCP      52162   15/12/87   0:00
```

```
4201          DCP     17081     15/12/87  0:00
5202          DCP       416     15/12/87  0:00

        4 File(s)   18245604 bytes free
```

Se viene specificato il pathname di una directory, DIR mostra l'elenco dei file in essa contenuti. Se viene specificata solo l'indicazione del drive, DIR mostra l'elenco dei file della directory corrente per quel drive.

Comandi correlati: COUNTRY in CONFIG.SYS

DISKCOMP

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando DISKCOMP confronta il contenuto di due dischi.

Sintassi DISKCOMP [drive:] [drive:]

Questo comando permette la comparazione fisica di due dischi. Se non viene specificato nessun drive, il confronto viene effettuato per il drive corrente: poiché non è possibile effettuare il confronto fra dischi rigidi, nel caso il nostro disco corrente sia C la digitazione del comando DISKCOMP senza alcun argomento provoca un messaggio di errore.
Per effettuare la comparazione di due dischi, dopo averli posti rispettivamente nei drive A e B, occorre digitare DISKCOMP A: B:. Se il nostro calcolatore ha un solo drive a dischi, per effettuare la comparazione di due dischi occorre digitare DISKCOMP A: A:.
Quando la comparazione avviene fra due dischi in un elaboratore dotato di un solo drive, il comando DISKCOMP richiede la sostituzione del primo disco da confrontare con il secondo, seguito dalla pressione del tasto ENTER per confermare l'avvenuta sostituzione, dopodiché, se il confronto non è terminato, la reimmissione del primo disco e così via fino al termine della comparazione. Una volta concluso positivamente il confronto, il messaggio:

```
Compare OK
```

segnala la riuscita, altrimenti, durante la comparazione, vengono segnalate le posizioni in cui i due dischi differiscono.
Il comando DISKCOMP può essere utilizzato solo con dischi del medesimo formato e capacità. Comunque termini la comparazione, DISKCOMP presenta la richiesta seguente:

```
Compare another diskette (Y/N)_
```

Rispondendo Y viene effettuata un'altra comparazione con la medesima specifica per ciò che riguarda le unità, rispondendo N il programma termina.

Comandi correlati: COMP, DISKCOPY

DISKCOPY

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando DISKCOPY copia l'intero contenuto di un disco su un altro.

Sintassi DISKCOPY [drive:] [drive:]

Questo comando permette la copia fisica di un disco. Se non viene specificato nessun drive, la copia viene effettuata utilizzando il drive corrente: poiché non è possibile effettuare copie fisiche fra dischi rigidi, nel caso il nostro disco corrente sia C la digitazione del comando DISKCOPY senza alcun argomento provoca un messaggio di errore.
Se il disco di destinazione non è stato preventivamente formattato (vedi comando FORMAT, più oltre), la formattazione viene eseguita durante la copia.
Per effettuare la copia, dopo avere posto il disco da copiare nel drive A ed un disco vuoto (o il cui contenuto non interessi più) nel drive B, occorre digitare DISKCOPY A: B:. Se il nostro calcolatore ha un solo drive a dischi, per effettuare la copia occorre digitare DISKCOPY A: A:.
Quando la copia avviene in un elaboratore dotato di un solo drive, il comando DISKCOPY richiede la sostituzione del disco sorgente con il disco destinazione seguito dalla pressione del tasto ENTER per confermare l'avvenuta sostituzione, dopodiché, se la copia non è terminata, la reimmissione del disco sorgente e così via fino al termine della copia. Una volta conclusa positivamente la copia, DISKCOPY conferma il successo dell'operazione segnalando il "numero di serie" attribuito al disco:

```
New Volume Serial Number on target drive is 1234-5678
```

DISKCOPY assegna un numero di serie ad ogni disco copiato, generandolo in modo univoco e realizzando così un identificatore univoco per tutti i nostri dischi.
Se, durante la copia, DISKCOPY rileva errori in uno dei dischi, segnala la posizione in cui è stato rilevato l'errore, proseguendo comunque la copia.
Il comando DISKCOPY può essere utilizzato solo con dischi del medesimo formato e capacità. Al termine della copia, DISKCOPY presenta la richiesta seguente

```
Copy another diskette (Y/N)_
```

Rispondendo Y viene effettuata un'altra copia con la medesima specifica per ciò che riguarda le unità, rispondendo N il programma termina.

Comandi correlati: COPY, XCOPY, DISKCOMP

DPATH

Ambiente: OS/2

Tipo: *Interno*

Il comando DPATH consente di definire percorsi di ricerca (path), all'interno del file system, per file non contenenti programmi.

Sintassi DPATH [{;|path [;path][...]}]

Con DPATH è possibile specificare una serie di una o più directory all'interno delle quali il sistema operativo ricerca i file che costituiscono, ad esempio, argomenti di altri comandi. Privo di argomenti, DPATH ci mostra l'elenco delle directory, o meglio dei path, precedentemente memorizzati separati da ";". Seguito dal solo ";" annulla qualsiasi path precedentemente memorizzato e infine, seguito da uno o più path, li memorizza, permettendone così l'uso al sistema operativo e ad altri comandi. Se, ad esempio, noi digitiamo DPATH immediatamente dopo l'accensione del calcolatore (attenzione, questo è un comando utilizzabile solo in modo protetto!) non vedremo apparire nulla sul monitor; se, successivamente digitiamo

```
DPATH C:\ARCHIVI;C:\RISERVAT\ARCHIVI
```

otterremo la memorizzazione di questi due path e, digitando nuovamente DPATH li vedremo comparire sul monitor. Se digitiamo DPATH ; ogni path verrà cancellato, cosa che possiamo verificare eseguendo nuovamente il comando.
A differenza del comando APPEND (vedi sopra) che effettua la medesima funzione per l'ambiente DOS, DPATH è un comando interno e non occorre alcuna specifica per far sì che le directory specificate entrino a far parte delle variabili di ambiente della sessione da cui abbiamo eseguito questo comando.

Comandi correlati: APPEND, PATH, SET

EXIT

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando EXIT permette di terminare il command processor corrente e di ritornare al command processor precedente o al program selector.

Sintassi EXIT

Il comando EXIT provoca il ritorno al command processor precedente: se il command processor era CMD, EXIT restituisce il controllo al comando da cui quest'ultimo era stato attivato (vedi sopra, CMD) o, se questo era il CMD con cui era stata attivato lo screen group, restituisce il controllo al program selector. Se il command processor era COMMAND, EXIT restituisce il controllo al comando da cui quest'ultimo era stato attivato (vedi sopra, COMMAND) o, se questo era il COMMAND con cui era stato attivato il "compatibility box" DOS, non fa nulla.
EXIT riveste un interesse particolare nel caso in cui, durante l'utilizzo di un programma applicativo, desideriamo eseguire dei comandi del sistema operativo senza interrompere il programma suddetto. Normalmente, il programma possiede un'opzione tramite cui richiamare un programma dal sistema operativo: richiamando il command processor, ci si presenta il prompt del sistema operativo e possiamo quindi eseguire i comandi desiderati; al termine, digitando EXIT, il command processor termina e il controllo viene restituito al programma da cui era stato avviato.

Comandi correlati: COMMAND, CMD

FDISK

Ambiente: OS/2

Tipo: *Esterno*

Il comando FDISK ha la funzione di configurare un disco rigido.

Sintassi FDISK

L'utility FDISK permette di suddividere un disco rigido in partizioni e specificare quale dev'essere il loro utilizzo. OS/2 ha un limite intrinseco nella dimensione massima di una singola unità: 32 Mbyte. Questo significa che per sfruttare appieno le odierne capacità dei dischi rigidi, che arrivano alle centinaia di megabyte, occorre suddividere questi dischi in tante unità logiche, nessuna delle quali deve avere dimensioni eccedenti i suddetti 32 Mbytes. In effetti, questo è il compito del comando FDISK.
La memorizzazione dei file sui dischi avviene su cerchi concentrici, presenti in ugual numero su ogni superficie del disco, detti tracce. Queste tracce sono numerate progressivamente a partire da 0, e il numero cresce avvicinandosi al perno centrale (asse di rotazione). Poiché i dischi rigidi sono costituiti da una serie di dischi ruotanti su un unico asse centrale, le tracce con il medesimo numero poste sulle diverse superfici individuano dei "cilindri virtuali": il metodo di suddivisione di un disco rigido in partizione consiste nell'indicazione di un intervallo di cilindri cui corrisponderà poi la lettera identificativa del drive.
Il comando FDISK consente di creare sul disco due diversi tipi di partizione: la partizione cosiddetta "primaria" o "principale" è quella dove risiede il sistema operativo e che può essere utilizzata per avviare il calcolatore; la partizione "estesa" o "secondaria" che può a sua volta essere suddivisa in "unità logiche" di massimo 32 Mbyte ciascuna.
La partizione principale, se presente, deve essere la prima; in altri termini l'intervallo di cilindri che individua la partizione principale deve contenere il cilindro 0. La partizione secondaria, se presente, può trovarsi sul medesimo disco o anche su un disco non dotato di partizione principale.
Il comando FDISK interagisce con l'utilizzatore con una serie di "menu" da cui l'utilizzatore stesso può scegliere l'operazione da compiere. Questo comando offre la possibilità di:

- Creare una partizione DOS.
- Cambiare la partizione attiva.
- Cancellare una partizione DOS.
- Visualizzare le informazioni relative ad una partizione DOS.
- Visualizzare o modificare la configurazione di un secondo disco rigido, nel caso in cui nel nostro calcolatore siano installati due dischi.

La prima opzione è normalmente attivata durante l'installazione (capitolo 8); la seconda opzione permette di segnalare al calcolatore che all'accensione, anziché partire con OS/2, dovrà partire con un diverso sistema operativo, preventivamente installato in una partizione libera del disco (un caso tipo è costituito dal sistema operativo Xenix: è frequente l'utilizzo dei personal computer con una partizione in DOS ed una partizione in Xenix). La terza opzione permette la cancellazione di una partizione con conseguente perdita dell'intero contenuto; la quarta opzione fornisce indicazioni sull'intervallo di cilindri in cui è definita una partizione e, se questa è una partizione secondaria, sulle unità logiche in essa definite. La quinta opzione, infine, è presente solo quando sul nostro calcolatore sono presenti due dischi rigidi: selezionandola apparirà un menu uguale a quello appena commentato con l'unica differenza che la quinta opzione permette di tornare al menu del primo disco.
Richiamando FDISK, compare la seguente maschera video (figura 5.1).

```
IBM Personal Computer
Fixed Disk Setup Program Version 1.00

FDISK options


Choose one of the following:

    1.  Create an IBM Operating System/2 partition
        or a logical drive
    2.  Change the active partition
    3.  Delete an IBM Operating System/2 partition
        or a logical drive
    4.  Display the partition data


Enter choice: [1]



Press Enter to continue or
Esc to return to IBM Operating System/2
```

Figura 5.1

Le cinque opzioni proposte corrispondono a quelle indicate sopra. sigla costruttore va letta come Microsoft o IBM, a seconda della provenienza della nostra versione di OS/2. Premendo il tasto ESC è possibile interrompere l'utility FDISK e ritornare al prompt del sistema operativo. Le altre scelte si possono selezionare

digitando il numero corrispondente all'opzione desiderata. Ogni volta che viene scelta un'opzione, viene presentato un valore di default: ad esempio, scegliendo l'opzione 1 per creare una partizione e successivamente l'opzione 2 per creare una partizione secondaria, viene mostrato il numero del primo cilindro libero e proposto il numero dell'ultimo cilindro utile del disco in maniera tale che, confermando questo valore, l'intera parte di disco non utilizzata per la partizione principale venga utilizzata per la partizione secondaria. Naturalmente noi possiamo voler ridurre questa partizione per, ad esempio, lasciare spazio ad una eventuale partizione Xenix: quello che faremo è di diminuire il numero proposto da FDISK come fine della partizione (facendo attenzione a lasciarlo comunque sufficientemente maggiore di quello iniziale, che non possiamo variare).
Occorre usare questo comando solo se è assolutamente necessario e con molta attenzione, in quanto le informazioni che esso gestisce riguardano il disco nella sua globalità: non a caso, dopo avere variato qualche cosa tramite il comando FDISK, il sistema si arresta e richiede di essere avviato nuovamente.

Attenzione! Occorre ricordare anche che, una volta creata una nuova partizione, per poterla utilizzare occorre formattarla (vedi comando FORMAT, più oltre).

FIND

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando FIND è un filtro che permette di visualizzare solo le linee di un determinato file che soddisfano certi criteri.

Sintassi FIND [/V] [/C] [/N] sequenza di caratteri [pathanme] [...]

FIND è un comando che, esaminando i file individuati dai pathname, ricerca in ciascuno di essi le linee che contengono sequenza di caratteri, cioè un certo numero di caratteri (tipicamente una o più parole), e le visualizza secondo le regole imposte tramite le tre opzioni, di seguito esposte:

/V	Visualizza le linee che *non* contengono la sequenza cercata
/C	Visualizza, per ciascun file, solo il numero totale delle linee che contengono la sequenza cercata. Se utilizzata insieme all'opzione /V, visualizza, per ciascun file, solo il numero totale delle linee che *non* contengono la sequenza cercata
/N	Premette ad ogni linea visualizzata il numero relativo alla posizione della linea nel file, assegnando il numero 1 alla prima linea di ogni file. Se utilizzata insieme all'opzione /C, questa opzione non viene presa in considerazione.

La sequenza di caratteri da ricercare deve essere racchiusa tra due coppie di virgolette (""). Se anche le virgolette costituiscono un carattere di ricerca, occorre racchiudere la sequenza desiderata tra due coppie di virgolette. FIND considera distinte parole uguali composte però da lettere rispettivamente maiuscole e minuscole. Negli eventuali pathname non è ammesso, per questo comando, l'uso dei caratteri "?" e "*".
Il comando FIND, quando non viene specificato alcun nome di file, ricerca la sequenza desiderata in STANDARD INPUT, cioè nelle righe digitate da tastiera dopo l'immissione del comando e fintantoché non viene digitato un carattere di fine file (ottenuto premendo contemporaneamente i tasti Ctrl e Z). Questa caratteristica è utile nella costruzione di filtri (capitolo 7).
Se, ad esempio, vogliamo verificare quante volte e in quali righe del nostro file di testo LETTERA.TXT compare la parola "attenzione", digiteremo:

```
FIND /N "attenzione" LETTERA.TXT
```

ottenendo:

```
— — — — — LETTERA.TXT
```

[5] sottoponiamo alla Vostra cortese attenzione il nostro listino
[12] Grati dell'attenzione che Vorrete prestare ai nostri prodotti

il che ci mostra che la parola cercata compare due volte, nella quinta e nella dodicesima linea del testo.

Comandi correlati: COMP

FORMAT

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando FORMAT permette di formattare un disco floppy o una partizione di un disco rigido.

Sintassi FORMAT drive [/4] [/S] [/T:numero tracce] [/N:numero settori] [/V[:nome volume]]

Il comando FORMAT prepara un disco, sia floppy che rigido, all'utilizzo da parte di OS/2. FORMAT, a seconda delle opzioni e del tipo di disco, predispone una serie di tracce concentriche su tutte le superfici di cui il disco è dotato suddividendo poi ciascuna di esse in un certo numero di settori, su cui i programmi andranno poi a memorizzare i vari file.
La formattazione (cioè il processo a cui il comando FORMAT sottopone un disco), dipende innanzitutto dalle caratteristiche del drive: i drive per floppy disk sono innanzitutto di due diversi misure: 5 pollici e 1/4 o 3 pollici e 1/2; inoltre, ciascuna di queste unità può essere o meno ad alta capacità, capaci pertanto di formattare un disco fino a 80 tracce e 18 settori per traccia.
Un comune drive da 5 pollici e 1/4 (quello di cui sono dotati la maggioranza dei personal computer PC compatibili) può formattare, e quindi utilizzare, un disco a 40 tracce, 9 settori per traccia per un totale di 360 Kbyte.
Un drive da 5 pollici e 1/4 ad alta capacità (quello di cui sono dotati i personal computer della classe AT) può formattare, e quindi utilizzare, un disco a 80 tracce, 15 settori per traccia per un totale di 1200 Kbyte. Questo drive può formattare, e quindi utilizzare, anche dischi da 360 Kbyte, tuttavia la formattazione eseguita su questi drive non è di grande affidamento se il disco viene poi utilizzato su drive comuni.
Un drive da 3 pollici e 1/2 può formattare, e quindi utilizzare, un disco a 80 tracce, 9 settori per traccia per un totale di 720 Kbyte.
Un drive da 3 pollici e 1/2 ad alta capacità (quello di cui sono dotati i nuovi Personal System/2 e simili) può formattare, e quindi utilizzare, un disco a 80 tracce, 18 settori per traccia per un totale di 1440 Kbytes. Anche in questo caso, questo drive può formattare, e utilizzare, dischi da 720 Kbyte. L'affidabilità è maggiore, sarebbe comunque preferibile eseguire almeno la formattazione con drive normali.
Drive indica il nome del drive dove si trova il disco da formattare; se si indica un disco rigido, il numero di richieste di conferma viene aumentato da parte del sistema operativo per evitare che venga formattato per errore il disco rigido, in quanto verrebbero perse tutte le informazioni in esso memorizzate. Le opzioni hanno il seguente significato:

/4	Effettua la formattazione di un disco da 360 Kbyte in un'unità da 5 pollici e 1/4 ad alta capacità;
/S	Al termine della formattazione, copia tutti i file il cui nome compare nel file FORMATS.TBL nel disco formattato: lo scopo di questa opzione è di ottenere un disco contenente una copia del sistema operativo al fine di poter avviare il calcolatore tramite questo disco. Per poter utilizzare questa opzione, occorre che il disco non abbia capacità inferiore a 1200 Kbyte, altrimenti non viene formattato;
/T:numero tracce	Questa opzione specifica il numero di tracce quando questo è diverso da quello del drive. Il valore di default è il massimo per ciascun drive;
/S:numero settori	Questa opzione specifica il numero di settori per traccia quando questo è diverso da quello del drive. Il valore di default è il massimo per ciascun drive;
/V:nome volume	Con questa opzione è possibile specificare un nome, valido per l'intero disco, che compare ogni volta che eseguiamo il comando DIR (vedi sopra). Se viene specificata l'opzione, ma non il nome, questo ci viene richiesto dal comando FORMAT al termine della formattazione.

Le uniche opzioni permesse anche per i dischi rigidi sono /S e /V; inoltre occorre ricordare che nei dischi rigidi ciò che viene formattato è la partizione principale o una unità logica contenuta nella partizione secondaria.
Durante la formattazione, un contatore ci tiene informati del numero di traccia e del numero di superficie in corso di formattazione; al termine, nel solo caso si tratti di un disco floppy, FORMAT chiede se vogliamo formattare un altro disco, visualizzando il messaggio:

```
Format another? (Y/N)
```

Rispondendo Y inizieremmo la formattazione di un altro disco, rispondendo N il controllo torna la sistema operativo.

Comandi correlati: LABEL, SYS, VOL

GRAFTABL

Ambiente: DOS

Tipo: *Esterno*

Il comando GRAFTABL permette di estendere il set di caratteri affinché possa essere utilizzato nel modo grafico.

Sintassi GRAFTABL [codice|?|/STA]

Codice è il numero identificativo del gruppo di caratteri che estendono il set standard del DOS. L'opzione "?" provoca la visualizzazione del nome del set caricato e di una lista delle opzioni di questo comando. L'opzione /STA provoca semplicemente la visualizzazione del nome del set caricato. Il comando GRAFTABL, digitato senza argomenti, provoca il caricamento del set di caratteri anglosassoni. I codici ammessi sono gli stessi del comando CODEPAGE in CONFIG.SYS (capitolo 9):

Codice	Nazione
437	Stati Uniti
860	Portogallo
863	Franco-canadese
865	Norvegese/Finlandese

HELP

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando HELP permette di visualizzare informazioni riguardo un messaggio d'errore.

Sintassi HELP [ON|OFF|codice]

Digitando solo HELP vengono visualizzate alcune regole fondamentali:

- come ritornare al program selector;
- come cambiare sessione (passare a un altro screen group);
- come terminare la sessione corrente;
- come ottenere informazioni aggiuntive su errori e messaggi generici.

Digitando HELP ON compare una linea che contiene informazioni sul tipo di sessione, su come ritornare al program selector e come attivare la funzione di "guida". Digitando HELP OFF la suddetta linea scompare.
Il codice è il numero che compare quando commettiamo un errore nella digitazione del nome di un comando (o per altre cause): il sistema operativo, ogni volta che visualizza un messaggio, premette al messaggio stesso una sigla di 3 lettere (SYS, DOS, ecc.), che individua l'ambiente a cui il messaggio si riferisce, seguita da un numero che lo identifica. Digitando HELP seguito da questo codice numerico, il sistema fornisce spiegazioni sulla causa probabile dell'errore e suggerimenti su come porvi rimedio.
Se il messaggio conteneva un riferimento ad un parametro da noi immesso, nei messaggi visualizzati dal sistema operativo verranno utilizzati i caratteri *** al posto di detto parametro.
Il file contenente i messaggi, che si chiama OSO001H.MSG, viene cercato dal sistema operativo nella directory corrente, dopodiché, se non è stato trovato, la ricerca, a seconda dell'ambiente, viene effettuata nelle directory specificate con il comando DPATH o APPEND (vedi sopra).

JOIN

Ambiente: DOS

Tipo: *Esterno*

Il comando JOIN permette di assegnare a un drive il nome di una directory di un altro drive.

Sintassi JOIN [drive: drive:\directory|drive: /D]

L'assegnazione di un drive ad una directory di un altro drive permette di accedere al primo drive tramite un path. Dopo l'esecuzione di un JOIN, non è più possibile accedere alla prima unità con la lettera identificativa, è possibile farlo solo tramite il path. JOIN, seguito dall'opzione /D, pone fine all'associazione relativa a drive, ripristinando la consueta simbologia a lettere per ciascun drive.
JOIN eseguito senza argomenti presenta l'elenco delle associazioni drive - directory in essere.
Il comando JOIN può avere come directory di assegnazione solo una directory posta al livello della directory root.
Se, ad esempio, il nostro drive corrente è C e noi vogliamo accedere al disco nel drive A tramite la directory DISCO A nel nostro drive, digiteremo:

```
JOIN A: C:\DISCOA
```

dopodiché, digitare CD \DISCOA sarà equivalente a cambiare in A il nostro drive corrente qualora il JOIN non fosse stato eseguito. In effetti, il comando DIR, eseguito dopo il precedente, mostrerebbe l'elenco dei file nella directory root del drive A. Digitando:

```
JOIN A: /D
```

cancelliamo l'assegnazione e ripristiniamo la possibilità di accedere al drive A tramite, appunto, la lettera identificativa.

Comandi correlati: SUBST, ASSIGN

KEYB

Ambiente: OS/2

Tipo: *Esterno*

Il comando KEYB permette di dichiarare il tipo di disposizione dei tasti della tastiera.

Sintassi KEYB sigla nazione

Sigla nazione può assumere i seguenti valori (i layout delle tastiere di ogni nazione sono riportati nell'appendice C):

Sigla	Nazione
BE	Belgio
CF	Canada (di lingua francese)
DK	Danimarca
SU	Finlandia
FR	Francia
GR	Germania
IT	Italia
LA	America latina
NL	Olanda
NO	Norvegia
PO	Portogallo
SP	Spagna
SV	Svezia
SF	Svizzera (di lingua francese)
SG	Svizzera (di lingua tedesca)
UK	Gran Bretagna
US	Stati Uniti

Il comando KEYB va eseguito in modo protetto, tuttavia ha effetto anche in emulazione DOS.
Per poter utilizzare il comando KEYB, occorre che nel file CONFIG.SYS (capitolo 9) sia stato inserito il comando DEVINFO per informare il sistema operativo della nostra intenzione di dichiarare la disposizione della tastiera.
Se in CONFIG.SYS è stato utilizzato anche il comando CODEPAGE (capitolo 9) occorre che i codici dei due set di caratteri dichiarati siano quelli validi per la nazione specificata, ancora nel file CONFIG.SYS, tramite il comando COUNTRY (capitolo 9). Questo perché il comando KEYB tenta di assegnare ai tasti della tastiera i valori del set di caratteri di default della nazione specificata: se questi

valori sono diversi perché con CODEPAGE sono stati dichiarati due set diversi da quelli della nazione individuata dalla sigla, KEYB presenta un messaggio di errore e chiede se vogliamo caricare comunque il set di caratteri di default per la nazione indicata o interrompere il comando.
Per predisporre la giusta interpretazione, da parte del calcolatore, dei tasti di una tastiera con il layout italiano, occorre aver inserito nel file CONFIG.SYS (capitolo 9) almeno queste due linee:

```
COUNTRY=039
DEVINFO=KBD, IT, KEYBOARD.DCP
```

(che stabiliscono di usare le convenzioni italiane per la rappresentazione di data, orario, ecc. e indicano il nome del file contenente la tabella di traslazione dei tasti della tastiera nei rispettivi caratteri italiani), dopodiché possiamo digitare:

```
KEYB IT
```

e otterremo quanto desiderato.

Comandi correlati: COUNTRY in CONFIG.SYS, DATE, TIME

LABEL

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando LABEL permette di creare o modificare il nome di volume dato a un disco.

Sintassi LABEL [drive:] [nome]

Il nome di volume può essere composto da 1 a 11 caratteri, esclusi i seguenti:

```
*?/\.,;:+=<>[]()&
```

Il nome di volume viene visualizzato ogni qual volta si eseguono i comandi CHKDSK e DIR (vedi sopra) o il comando VOL (vedi oltre).
Drive indica il drive contenente il disco di cui si vuole cambiare il nome: se omesso, viene preso in considerazione il drive corrente. Se nome non viene specificato, LABEL mostra quello corrente e richiede l'immissione del nuovo nome. Se, ad esempio, digitiamo:

```
LABEL C:
```

otterremo:

```
Volume in drive C is OS2
Type a volume label of up 11 characters
or press Enter for no volume label update:
```

cioè la specifica del nome attuale del disco (OS2) e la richiesta di inserire un nuovo nome, lungo fino ad 11 caratteri.

Comandi correlati: FORMAT, VOL

MKDIR

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando MKDIR crea una nuova directory.

Sintassi

```
MKDIR pathname [...]
MD pathname [...]
```

Questo è il comando che permette di costruire la struttura del file system (capitolo 3) su un disco: appena formattato (vedi sopra, comando FORMAT), un disco contiene solo la directory root ed è completamente vuoto (a meno che non sia stato formattato con l'opzione /S, nel qual caso contiene i file del sistema operativo).Il comando MKDIR, che può essere abbreviato MD, permette di creare delle directory, inizialmente vuote. Se, ad esempio, volessimo creare la struttura dell'esempio del capitolo 3, supponendo di trovarci nella directory root del disco C, dovremmo digitare:

```
MKDIR CONDOMIN
MKDIR RISTRUTT
MKDIR CONDOMIN\VIA_BARI
MKDIR CONDOMIN\VIA_ROMA
MKDIR CONDOMIN\VIA_ROMA\ROSSI
MKDIR CONDOMIN\VIA_ROMA\BIANCHI
```

Solo in emulazione DOS è obbligatorio fornire un solo pathname per volta al comando MKDIR; in modo protetto è possibile effettuare la stessa operazione richiamando il comando una sola volta, ponendo comunque attenzione a non creare una directory "contenuta" in un'altra senza aver prima creato quest'ultima:

```
MKDIR CONDOMIN RISTRUTT CONDOMIN\VIA_BARI CONDOMIN\VIA_ROMA
    CONDOMIN\VIA_ROMA\ROSSI CONDOMIN\VIA_ROMA\BIANCHI
```

Comandi correlati: CHDIR, RMDIR, DIR, TREE

MODE

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando MODE definisce alcuni parametri riguardo le porte seriali, le stampanti e il monitor.

Sintassi per le stampanti

```
MODE {PRN|LPTn} [numero caratteri][,numero linee][,P]
```

Con questa sintassi vengono definiti i parametri per le stampanti parallele connesse come LPT1 (equivalente a PRN), LPT2 e LPT3 (n individua appunto il numero 1, 2 o 3). Numero caratteri specifica qual è il numero massimo di caratteri che vanno scritti per ogni riga: i valori possibili sono 80 e 132, il valore di default è 80. Numero linee specifica la densità di stampa delle linee, che si misura in linee per pollice: i valori possibili sono 6 e 8, il valore di default è 6. L'opzione P provoca la ripetizione all'infinito del tentativo di mandare alla stampante i dati da stampare in caso di errore di sincronizzazione. I parametri sono posizionali, pertanto, se vogliamo specificare solo uno di essi, dobbiamo indicare l'assenza di quelli che eventualmente lo precedono con una virgola.
Se, ad esempio, la stampante che abbiamo collegato a LPT1 permette la stampa di 132 caratteri, digiteremo:

```
MODE LPT1 132 (o MODE PRN 132)
```

se, invece, la stampante che abbiamo collegato a LPT2 necessita del parametro P, digiteremo:

```
MODE LPT2 ,,P
```

Sintassi per le porte seriali in emulazione DOS

```
MODE COMn:baud[,parità][,databits][,stopbits][,P]
```

Sintassi per le porte seriali in modo protetto

```
MODE COMn[:][baud][,parità][,databits][,stopbits]
        [,TO={ON|OFF}][,XON={ON|OFF}][,IDSR={ON|OFF}][,ODSR={ON|OFF}]
        [,OCTS={ON|OFF}][,DTR={ON|OFF|HS}][,RTS={ON|OFF|HS|TOG}]
```

I primi quattro parametri sono uguali per le due modalità operative: per entrambe,

inoltre, sono posizionali, pertanto la specifica di uno solo di essi va effettuata sostituendo la virgola a quelli omessi che eventualmente lo precedono.

n — È il numero della porta seriale cui si riferiscono gli altri parametri, e può assumere valore 1, 2 o 3 (quest'ultimo valore solo per i personal computer della classe Personal System/2).

baud — Definisce la velocità di trasmissione: un baud equivale a un bit al secondo (il bit è l'unità di misura elementare con cui operano tutti gli elaboratori: può assumere soltanto i valori 0 e 1 e corrisponde alla variazione di tensione elettrica all'interno della CPU). Questo parametro può assumere i seguenti valori: 110, 150, 300, 600, 1200, 2400, 4800, 9600 e 19200. È sufficiente specificare le prime due cifre (le prime tre per 19200); il valore di default è 1200.

parità — La parità è un metodo elementare per il controllo dell'esattezza del byte ricevuto: poiché un byte è costituito da un certo numero di bit e ciascuno di essi può avere indifferentemente valore 0 o 1, si aggiunge un bit per far sì che il numero totale di tutti i bit a 1 per quel byte sia pari o dispari. In effetti questo parametro può assumere i seguenti valori: N, cioè nessun controllo di parità e quindi nessun bit aggiuntivo; O, cioè parità dispari quindi il bit aggiuntivo avrà valore tale per cui il numero totale di bit a 1 sia dispari; E, cioè parità pari (questo è anche il valore di default) quindi il bit aggiuntivo avrà valore tale per cui il numero totale di bit a 1 sia pari; M, per obbligare l'inserimento del bit di parità con valore 1; S, per obbligare l'inserimento del bit di parità con valore 0.

databits — Questo parametro indica il numero di bit che vanno considerati significativi, in altri termini di quanti bit è costituito il byte (il byte è stato definito nel capitolo 2 come lo spazio minimo necessario al calcolatore per memorizzare un carattere: esistono diverse codifiche che assegnano a una certa combinazione di bit un determinato carattere, quindi il numero di caratteri rappresentabili dipende dal numero di bit utilizzati per rappresentare ciascuno di essi). Questo parametro può assumere i valori 5,6,7 e 8 e il valore di default è 7.

stopbits — Questo parametro specifica il numero di bit che determinano dove finisce il byte. Può assumere i valori 1, 1,5 e 2: il valore 1,5 è valido soltanto quando il numero di databits è uguale a 5. Il valore di default è 2 quando baud vale 110, altrimenti è 1.

L'opzione P è valida soltanto in modo reale ed ha il medesimo effetto definito sopra per le porte parallele. In pratica, si utilizza questa opzione quando la porta seriale viene utilizzata per il collegamento di una stampante perché provoca la ripetizione all'infinito del tentativo di mandare alla stampante i dati da stampare in caso di errore di sincronizzazione.
I parametri che seguono valgono soltanto in modo protetto, non sono posizionali, ma vanno comunque inseriti dopo i quattro precedentemente esposti:

TO= Questo parametro riveste, per il modo protetto, la stessa funzione del parametro P per il modo reale: posto ad ON, provoca la ripetizione all'infinito della trasmissione dei dati in caso di errore di sincronizzazione con l'unità di destinazione. Il valore di default è OFF.

XON= Se posto a ON, questo parametro abilita il controllo del flusso di trasmissione da parte della porta seriale, se posto ad OFF, che è il valore di default, lo disabilita. Per "controllo del flusso di trasmissione" s'intende l'abilità da parte del calcolatore di interrompere momentaneamente la trasmissione dei dati quando riceve dall'unità di destinazione un carattere particolare, detto XOFF, e di riprenderla quando riceve dall'unità di destinazione un altro carattere particolare detto, appunto, XON. Questo tipo di controllo, che in anglosassone si chiama "handshake", è detto "software handshake" perché realizzato tramite caratteri trasmessi sul medesimo canale su cui viaggiano le altre informazioni, in contrapposizione all'"hardware handshake", realizzato tramite segnali elettrici che viaggiano su appositi fili. Questo controllo è utile quando l'unità di destinazione, ad esempio una stampante, ha bisogno di un certo tempo per elaborare i dati ricevuti prima di poterne ricevere altri.

IDSR= Questo parametro abilita o disabilita l'"hardware handshake" effettuato, per quanto riguarda i dati ricevuti dal calcolatore, tramite il filo detto RTS. Il valore ON, che è il valore di default, abilita questo controllo, il valore OFF lo disabilita.

ODSR= Questo parametro è l'analogo del precedente per quanto riguarda la trasmissione dei dati da parte dell'elaboratore. Il valore ON, che è il valore di default, abilita questo controllo, il valore OFF lo disabilita. OCTS= Questo parametro abilita o disabilita l'"hardware handshake" della trasmissione di dati da parte dell'elaboratore utilizzando il filo detto CTS. Il valore ON, che è il valore di default, abilita questo controllo, il valore OFF lo disabilita.

DTR= Il valore ON di questo parametro, che è il valore di default,

attiva il segnale sul filo detto DTR: questo segnale è spesso utilizzato dall'unità connessa per verificare la "presenza" del calcolatore, cioè, in altri termini, che il calcolatore sia acceso. Il valore OFF elimina il segnale da detto filo e, infine, il valore HS permette di effettuare l'"hardware handshake" tramite questo filo.

RTS= Come per il comando precedente, il valore ON, che è il valore di default, attiva il segnale sul filo detto RTS: anche questo segnale costituisce una sorta di indicazione di "pronto" che il calcolatore fornisce all'unità connessa. Il valore OFF elimina il segnale da questo filo e il valore HS permette di effettuare l'"hardware handshake" tramite questo filo. Il valore TOG, infine, lega la presenza e l'assenza di segnale sul filo RTS allo stato di altri parametri hardware.

In modo protetto, è possibile utilizzare il comando MODE con il solo nome della porta per visualizzare i valori dei parametri; in modo reale il comando MODE può essere utilizzato solo per alterare i valori della porta.
Dovendo collegare una stampante alla porta seriale 1, e questa stampante funzioni a 9600 baud, nessun controllo di parità, 8 databits, 1 stopbit e il "software handshake", digiteremo:

```
MODE COM1:96,N,8,XON=ON
```

per il modo protetto, mentre in emulazione DOS digiteremo:

```
MODE COM1:96,N,8
```

e dovremo realizzare il controllo di trasmissione all'interno del programma di stampa, in quanto in modo reale il controllo del flusso di trasmissione tramite i caratteri XON e XOFF è sempre disabilitato.
In entrambi gli ambienti, affinché MODE ottenga l'effetto desiderato sulle caratteristiche di trasmissione della porta seriale, occorre che sia stato specificato il comando DEVICE con i parametri riguardanti le porte seriali nel file CONFIG.SYS (capitolo 9).

Sintassi per i monitor

```
MODE {40|80|BW40|BW80|CO40|CO80|MONO}[,numero linee]
```

Il parametro che segue il comando MODE specifica le caratteristiche del monitor: 40 o 80 stabiliscono il numero di caratteri per linea che il monitor è in grado di visualizzare. Quando BW viene premesso ai due numeri suddetti, si specifica che il monitor è in bianco e nero, quando viene premesso CO si specifica che il monitor

è a colori. Se invece viene specificato MONO, si intende un monitor monocromatico comunque in grado di visualizzare 80 caratteri per linea.
Il parametro numero linee specifica il numero massimo di linee che detto monitor può visualizzare. I valori ammessi sono 25, 43 e 50 e il valore di default è 25.
Una importante differenza del comando MODE eseguito per i monitor è data dal fatto che le specifiche, in questo caso, riguardano solo la sessione corrente, contrariamente alle precedenti che riguardano tutto il sistema.

MORE

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando MORE permette di visualizzare i dati che un generico programma presenta sul video una schermata per volta.

Sintassi MORE (dati provenienti da STANDARD INPUT)

Il comando MORE permette di fare con altri programmi quello che l'opzione /P fa per il comando DIR (vedi sopra), cioè evitare che una serie di informazioni fornite da un generico programma scorrano troppo rapidamente sul video, tanto da impedirci di prenderne visione prima di scomparire. MORE provvede a interrompere questo scorrimento al riempimento del video, dopodiché si arresta presentando sull'ultima riga del monitor il messaggio:

```
— More —
```

e, alla pressione di qualsiasi tasto, prosegue per un'altra schermata.
Il comando MORE non ha argomenti, e riceve i dati che poi mostra tramite lo STANDARD INPUT, cioè il "canale" di comunicazione a cui è solitamente collegata la tastiera. Poiché non ha evidentemente molto senso digitare qualcosa al fine di riempire tutte le linee dello schermo per vedere MORE in azione, si intuisce che vi deve essere un modo per far sì che i dati visualizzati da altri programmi vengano incanalati nello STANDARD INPUT di MORE: questo si ottiene con l'utilizzo del segno "|" quando i dati da visualizzare tramite MORE provengono da un programma, e con l'utilizzo del segno "<" quando i dati da visualizzare provengono da un file.
In altri termini, se avessimo un file di testo molto lungo, chiamato LETTERA.TXT e posto nella nostra directory corrente, e volessimo visualizzarne il contenuto una schermata per volta, dovremmo digitare:

```
MORE <LETTERA.TXT
```

Se, viceversa, volessimo evitare che le prime differenze risultanti dalla comparazione di due file diversi, PROVE.MAT e PROVE.FER entrambi posti nella nostra directory corrente, sfuggano troppo velocemente verso l'alto, dovremmo digitare:

```
COMP PROVE.MAT PROVE.FER | MORE
```

I caratteri "|" e "<" che vanno, insieme ad altri, nel capitolo più generale del "reindirizzamento" dell'Input/Output, verranno trattati in dettaglio al capitolo 7.

PATCH

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando PATCH permette di modificare uno o più byte all'interno di un file.

Sintassi PATCH pathname [/A]

Questo comando permette di intervenire direttamente sul contenuto di un file, ed è utile in special modo se detto file non è un file di testo (per modificare il quale esistono programmi più semplici, come EDLIN che vedremo al capitolo 6). Se, ad esempio, in uno dei file che contengono i comandi esterni di OS/2 vi fosse un banale errore, anziché spedire un disco con la versione del programma corretta a tutti gli acquirenti, i distributori potrebbero mandare semplicemente una circolare che spieghi qual è (o quali sono) il byte da cambiare con il comando PATCH per correggere il malfunzionamento.
Se l'opzione /A non è presente, pathname individua il file in cui si vogliono effettuare i patch: in questo caso, le variazioni vanno fatte dall'utilizzatore. Nel caso vi sia l'opzione /A, pathname individua un file, solitamente fornito da un distributore, che contiene sia i nomi dei file da variare che le correzioni da effettuare: in questo caso, il procedimento è automatico e, salvo il sopravvenire di errori durante l'esecuzione del patch, l'utilizzatore non deve fare alcunché.
Nel caso di un patch manuale, il primo dato che occorre fornire è la posizione del byte da cui iniziare la variazione ("offset" in anglosassone), misurata come distanza in byte dal primo byte del file e inserita come numero esadecimale. A questo punto, il comando patch presenta i valori di 16 byte a partire dalla posizione indicata: ciascuno di essi può essere variato digitando un nuovo valore, sempre in esadecimale, che prenderà il posto di quello mostrato, oppure lasciato invariato premendo la barra spaziatrice. Dopo avere confermato o variato il sedicesimo byte, vengono presentati i 16 successivi e così via. In qualsiasi momento, premendo ENTER, ci viene chiesto se vogliamo effettuare altri patch: rispondendo Y ci viene richiesto un altro offset e si ripete quanto già visto, rispondendo N ci vengono mostrati tutti i patch effettuati e la richiesta di renderli effettivi. Rispondendo Y, tutte le modifiche digitate saranno effettivamente riportate sul file identificato dal pathname alle posizioni specificate, rispondendo N il file rimane invariato.
Unica precauzione da prendere quando intendiamo usare il patch è quindi fare prima una copia del file originale, così da poterlo recuperare nel caso in cui, una volta effettuati i patch, il programma non funzioni più.

PATH

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando PATH consente di definire percorsi di ricerca (path), all'interno del file system, per i comandi esterni e i programmi in generale.

Sintassi PATH [;|path[;path][...]]

Grazie a questo comando, è possibile eseguire programmi senza che i file che li contengono si trovino necessariamente nella directory corrente di chi nerichiede l'esecuzione. Come già visto nel capitolo 4, alla digitazione di un comando al prompt del sistema operativo corrisponde da parte di questo una ricerca del comando stesso, innanzitutto fra i comandi interni, nel qual caso detto comando viene eseguito senza ulteriori ricerche, altrimenti vengono ricercati nell'ordine file con il medesimo nome di quello digitato e con estensione .COM, .EXE e, a seconda dell'ambiente, .CMD o .BAT. Se questa ricerca risulta infruttuosa nella directory corrente, allora il sistema operativo effettua la ricerca nelle directory specificate tramite il comando PATH, se ve ne sono, ripetendo, per ciascuna di esse, lo stesso tipo di ricerca effettuato per quella corrente.
Digitando semplicemente PATH ci viene presentato l'elenco dei path già dichiarati, separati l'uno dall'altro dal ";".
Digitando PATH ; tutti i path precedentemente specificati vengono annullati: questo comando equivale a limitare la ricerca dei programmi nella directory corrente di chi ne richiede l'esecuzione o a fornirne il pathname assoluto.
Infine, digitando PATH seguito da uno o più pathname separati dal ";", i path specificati costituiscono le directory in cui il sistema operativo effettuerà la ricerca dei comandi digitati.

Comandi correlati: APPEND, DPATH, SET

PRINT

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando PRINT stampa un file o lo toglie dalla coda di stampa.

Sintassi PRINT [/D:uscita] [/T|pathname[/B|/C] [...]]

PRINT permette di stampare file tramite il dispositivo connesso a uscita: LPT1, che equivale a PRN, è il valore di default quando l'opzione /D non è specificata. Se il comando SPOOL (vedi oltre) non è stato attivato, PRINT stampa il file richiesto impegnando direttamente l'uscita specificata e, di conseguenza, la sessione corrente finché la stampa non è terminata. Inoltre, se detta uscita è occupata da un'altra stampa, avviata da un altro programma, PRINT attenderà la fine di quest'ultima. Se invece il comando SPOOL è stato attivato, PRINT non fa altro che girare le richieste di stampa che gli pervengono a questo comando; il quale, ogni volta che eseguiamo PRINT seguito dal pathname del file (anche più di uno) che vogliamo stampare, aggiorna una lista di file in attesa di stampa che esaurisce gradualmente; questo metodo di gestire le richieste, detto "coda di stampa", è molto utile nell'ambiente multitask perché permette di disimpegnare immediatamente la sessione che ha avviato la stampa affinché possa essere dedicata ad altre elaborazioni, mentre la stampa viene effettuata senza impegnare nessun altro screen group.
PRINT non può essere digitato da solo, tuttavia nessun parametro è obbligatorio: se è specificata l'uscita non è indispendabile che vi sia alcun pathname in quanto può semplicemente essere specificata l'opzione /T che cancella la coda di stampa per il dispositivo specificato, vale a dire annulla tutte le stampe in attesa di esecuzione. L'opzione /C, invece, cancella dalla coda di stampa i file individuati da pathname: non richiede che l'uscita sia specificata, nel qual caso cancella i file suddetti dalla coda di stampa della stampante LPT1 altrimenti dalla coda di stampa del dispositivo specificato. L'opzione /B specifica che il file individuato da pathname potrebbere contenere dei caratteri di fine file, ma va ugualmente stampato per tutta la sua lunghezza. Il caso più semplice non richiede alcuna opzione: solo PRINT seguito dal nome del file da stampare sulla stampante collegata a LPT1.
È evidente che i parametri /C e /T richiedono che il programma SPOOL sia stato attivato.

Comandi correlati: SPOOL

PROMPT

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando PROMPT modifica il prompt del sistema operativo.

Sintassi PROMPT [nuovo prompt]

Il prompt del sistema operativo può essere cambiato ed essere adattato a particolari necessità. Qualsiasi testo può essere inserito come nuovo prompt, inoltre è possibile utilizzare le sigle seguenti:

Sigla	Significato
$$	Il carattere "$".
$_	A capo (cioè il prompt vero e proprio si posizionerà alla riga seguente).
$a	Il carattere "&".
$b	Il carattere "|".
$c	Il carattere "(".
$d	La data del sistema.
$e	Il carattere Escape (cioè quello che otterremmo premendo il tasto ESC).
$f	Il carattere ")".
$g	Il carattere ">".
$h	Il carattere Backspace (cioè quello che otterremmo premendo il tasto ←).
$i	La linea di guida (quella che otterremmo con il comando HELP ON).
$l	Il carattere "<".
$n	Il nome del drive corrente.
$p	Il pathname assoluto della nostra directory corrente.
$q	Il carattere "=".
$s	Lo spazio, valevole soltanto come primo carattere del nuovo prompt.
$t	L'orario del sistema.
$v	Numero della versione di OS/2.

Il prompt dei due diversi ambienti è definito di default dal sistema operativo in due modi diversi: per il modo reale è definito come lettera identificativa del drive corrente seguita dal carattere ">"; per il modo protetto è definito come pathname assoluto della nostra directory corrente racchiuso fra i caratteri "[" e "]". Digitando PROMPT senza alcun argomento viene ripristinato il prompt di default dell'ambiente in cui ci si trova. Il comando PROMPT agisce soltanto sulla sessione

in corso. Se, ad esempio, volessimo ottenere un prompt con orario e pathname della nostra directory corrente, dovremmo digitare:

```
PROMPT $p;$t$h$h$h$h$h$h$g
```

e, supponendo che siano le 18 e 30 e di trovarci nella directory root del drive C, otterremmo:

```
C:\;18:30>
```

Il ";" è stato messo per distingue i due argomenti, i sei caratteri "backspace" per togliere dall'indicazione dell'orario i secondi e i centesimi di secondo, altrimenti visualizzati dalla sigla $t.

Comandi correlati: DATE, TIME, VER

RECOVER

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando RECOVER permette di recuperare file che contengono settori di disco danneggiati.

Sintassi RECOVER {drive:|pathname}

Se, durante il normale utilizzo del calcolatore, alcuni settori di un disco (tipicamente di un disco rigido), dovessero danneggiarsi, i file il cui contenuto si trovava, in tutto o in parte, nei settori rovinati, non sono più utilizzabili. La possibilità che uno o più settori di un disco si rovinino (o meglio, perdano la capacità di mantenere memorizzati i dati che vi sono stati registrati) non è da trascurare, anche se molto rara. Nel caso ciò accada, se è noto il nome del file contenente il settore rovinato, tramite il comando RECOVER seguito dal pathname di detto file siamo in grado di recuperare almeno le parti che non si trovavano nel suddetto settore, settore che inoltre viene segnato da OS/2 per non essere più utilizzato. Se non è chiaramente individuabile il file contenente il settore rovinato o se sia file che settori sono più di 1, si può utilizzare il comando RECOVER seguito dalla lettera identificativa del drive: OS/2 verificherà allora tutti i settori del disco segnando tutti quelli trovati difettosi affinché non vengano più utilizzati e tentando di recuperare i file memorizzati in questi settori, salvo per la parte riguardante i settori medesimi. Ad esempio, per effettuare verifica ed eventuale recupero di file per sospetti settori danneggiati su un disco posto nel drive A, digiteremo:

```
RECOVER A:
```

Comandi correlati: CHKDSK

RENAME

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando RENAME permette di cambiare il nome di un file.

Sintassi

```
RENAME pathname nome
REN pathname nome
```

RENAME cambia il nome del file individuato da pathname in nome, per il quale non occorre specificare alcun path in quanto RENAME non può cambiare la directory in cui si trova un file ma anzi, utilizza il path fornito con il primo file per posizionarsi nella directory specificata, quindi cambiare nome al file suddetto. REN è un'abbreviazione di RENAME che funzione alla stessa maniera.
Nella directory in cui si trova il file a cui si vuole cambiare nome non deve esistere alcun altro file con il nome che si intende dare al primo, altrimenti il comando RENAME non avrà successo e presenterà un messaggio di errore.
Sia pathname che nome possono contenere i caratteri "?" e "*"; tuttavia occorre digitare il comando RENAME in modo che non vi siano possibilità di ambiguità fra i nomi che vengono cambiati e i nuovi nomi. Se, ad esempio, volessimo cambiare nome a tutti i nostri file di testo, posti nella directory APPUNTI del drive corrente, aggiungendo al nome di ciascuno di essi il carattere A iniziale, dovremmo digitare:

```
REN \APPUNTI\*.TXT A*.TXT
```

mentre è un errore digitare, ad esempio, REN \APPUNTI*.TXT APP.TXT quando nella directory appunti esiste più di un file con estensione TXT: in effetti con quest'ultimo comando RENAME dovrebbe cambiare nome a più di un file pur avendo un solo nome nuovo, e questo dà origine ad un messaggio di errore.

Comandi correlati: COPY

REPLACE

Ambiente: *Entrambi*

Tipo: *Esterno*

il comando REPLACE permette di sostituire file con versioni aggiornate e di aggiungere nuovi file.

Sintassi REPLACE pathname [path] [/A|/S] [/P] [/R] [/W]

Questo comando permette la sostituzione dei file individuati da pathname con altri, localizzati da path ed aventi il medesimo nome. Le opzioni hanno il significato seguente:

/A	Copia i file specificati con pathname solo se non esistono in path: in questo modo è possibile aggiungere file da un disco ad un'altro senza sovrascrivere quelli con nomi uguali.
/S	Ricerca i file specificati da pathname in tutte le directory a partire da path: normalmente, in questo caso, path individua la directory root del disco in cui effettuare le sostituzioni.
/P	Questa opzione provoca una richiesta di conferma per ogni file che, in base alle specifiche precedenti, risulti da sostituire o da aggiungere, richiesta cui è possibile rispondere Y o N per effettuare o meno la sostituzione/aggiunta, dopodiché REPLACE passa al file seguente, ripetendo la richiesta e così via.
/R	Estende l'operazione di sostituzione ai file che sono dichiarati di sola lettura (vedi comando ATTRIB, più sopra): in assenza di questa opzione, un file con questo attributo non viene sostituito.
/W	Richiede una conferma prima di iniziare ad effettuare la sostituzione: questo per permettere, ad esempio, di inserire un disco, contenente i file da sostituire, in un drive. L'esecuzione del comando parte premendo qualsiasi tasto.

Se, ad esempio, ricevessimo un disco contenente una nuova versione di alcuni comandi posti, nel nostro disco C, in diverse directory, per effettuare la sostituzione dovremmo digitare:

```
REPLACE A:\*.* C:\ /S
```

Se, inoltre, fossimo a conoscenza che il disco contiene alcuni nuovi comandi, dovremmo utilizzare l'opzione /A. Supponendo nel frattempo, di avere cambiato

la nostra directory corrente nella directory OS2 del drive C, dove risiedono i comandi, per aggiungere questi nuovi file dovremmo digitare:

```
REPLACE A:\*.* /A
```

e i comandi verrebbero aggiunti nella nostra directory corrente.

RESTORE

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando RESTORE ripristina i file memorizzati tramite il comando BACKUP.

Sintassi RESTORE drive sorgente: drive destinazione:[pathname] [/P][/M] [/S][/N][/B:[data]] [/A:data] [/E:ora] [/L:ora]

Tramite il comando RESTORE è possibile ripristinare, in modo selettivo sul drive destinazione, i file individuati da pathname memorizzati tramite precedenti BACKUP (vedi sopra) su drive sorgente. Il pathname che individua i file da ripristinare può, ovviamente, contenere i caratteri "?" e "*" per permettere il ripristino per categorie omogenee. Il significato delle opzioni è il seguente:

/P Questa opzione provoca una richiesta di conferma quando il ripristino riguarda un file che esiste su drive destinazione ed è variato rispetto a quello salvato o è di sola lettura: rispondendo Y il file viene ripristinato dal drive sorgente, rispondendo N il file non viene ripristinato; in entrambi i casi il ripristino continua con il file successivo.

/M Questa opzione provoca il ripristino dei soli file che abbiano subito qualche variazione dall'ultimo salvataggio effettuato tramite BACKUP.

/S Se questa opzione non è specificata e vi sono, nel drive sorgente, più livelli di directory, vengono ripristinati solo i file presenti al livello più alto. In presenza di questa opzione, invece, tutte le eventuali sottodirectory con i relativi file vengono ripristinate.

/N Ripristina i file memorizzati su drive sorgente solo se non esistono su drive destinazione.

/B:data Ripristina solo i file la cui data di creazione o di ultima modifica su drive destinazione risulti uguale o minore a data. La data, analogamente al comando BACKUP, va specificata nel formato stabilito dal comando COUNTRY (nazione, capitolo 9) che, nel caso dell'Italia, è GG/MM/AA (cioè due cifre per il giorno, 2 per il mese e le ultime 2 dell'anno).

/A:data Analogamente all'opzione precedente, ripristina solo i file la cui data di creazione o di ultima modifica su drive destinazione risulti uguale o maggiore a data, cui si applicano le medesime considerazioni fatte nel trattare l'opzione /B.

/E:ora In questo caso, il parametro da prendere in considerazione

è l'orario di creazione o di ultima modifica dei file: verranno ripristinati solo i file il cui orario di creazione o di ultima modifica su drive destinazione risulta uguale o minore di ora. L'orario va sempre specificato nel formato OO:MM:SS (cioè due cifre per l'ora, 2 per i minuti primi e 2 per i minuti secondi).

/L:ora — Questa opzione agisce esattamente come la precedente, salvo che i file ripristinati saranno solo quelli il cui orario di creazione o di ultima modifica su drive destinazione risulta uguale o maggiore di ora, sempre specificata nel formato suddetto.

Il comando RESTORE è in grado di effettuare ripristini da tutti i tipi di memoria di massa che il calcolatore può gestire: questo risulta evidente anche dall'affermazione, fatta più sopra a proposito del comando BACKUP, che quest'ultimo comando è in grado di effettuare le copie di salvataggio sullo stesso spettro di unità ed è evidente che non sarebbe di alcuna utilità un comando che salva qualcosa per cui non esiste il comando di ripristino.
Ad esempio, il comando per ripristinare quanto abbiamo salvato con l'esempio fatto per il comando BACKUP, è:

```
RESTORE A: C: /S
```

Se, invece, ad esempio, volessimo ripristinare solo i file che nel frattempo sono cambiati per riavere la situazione esattamente come era al momento del salvataggio, dovremmo consultare il file SALVA, che contiene la data e l'ora di quel salvataggio, che supporremo essere stato fatto il 29 gennaio alle 15,30, e digitare:

```
RESTORE A: C: /S /A:29/01/88 /L:15:30:00
```

Otterremo così il risultato richiesto. Il comando BACKUP richiede l'immissione del disco 01 nel drive sorgente, ed inizia il ripristino: se il disco inserito non è quello richiesto o porta un altro numero di sequenza, BACKUP visualizza un messaggio di errore e ripete la richiesta, altrimenti estrae i file secondo i parametri specificati e richiede l'inserimento del disco successivo e così via fino al termine del salvataggio.

Comandi correlati: ATTRIB, BACKUP

RMDIR

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando RMDIR elimina una directory.

Sintassi
RMDIR pathname [pathname] [...]
RD pathname [pathname] [...]

L'eliminazione di una directory è possibile solo se essa è vuota; notare che una directory viene considerata vuota quando contiene soltanto "." e ".." che rappresentano rispettivamente la directory stessa e la directory padre, come abbiamo visto nel capitolo 3. RD è un'abbreviazione di RMDIR che funziona nella stessa maniera.
Non è possibile cancellare, anche se vuote, la nostra directory corrente e la directory root di qualsiasi disco.
Per eliminare una directory occorre individuarla tramite pathname: in modo reale è possibile cancellare un sola directory per volta, in modo protetto, invece, è possibile specificare una serie di directory da cancellare tramite una sola esecuzione di RMDIR.
È possibile che in una directory si trovino dei file nascosti: non è possibile con gli strumenti forniti con OS/2 verificare la presenza di tali file; il comando CHKDSK, che abbiamo già visto, fornisce un dato riassuntivo sui file nascosti presenti su di un disco, senza specificare però dove essi si trovino. Tuttavia, se il comando RMDIR si rifiuta di cancellare una directory, visualizzando un messaggio di errore da cui si deduce che la directory non è vuota, anche se sembra esserlo, e CHKDSK riporta un numero eccessivo di file nascosti (3 sono comunque parte del sistema operativo), si deduce automaticamente che almeno uno di essi si trova nella directory che non riusciamo a cancellare. Attualmente, esistono diversi programmi, come ad esempio PC Tools e Norton Utilities, che permettono di manipolare i file nascosti: sono, naturalmente, utilizzabili solo in emulazione DOS.

Comandi correlati: CHDIR, MKDIR, DIR, TREE

SET

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando SET visualizza o modifica le variabili di ambiente.

Sintassi SET [nome variabile=[valore]]

A ogni screen group (o sessione, vedi capitolo 4), attivato dal program selector, viene associato un "ambiente": parte di questo ambiente ci risulta completamente indifferente, anche se, probabilmente, è la parte più importante, quella cioè che riguarda il fatto che la zona di memoria assegnata a ciascuna sessione è indipendente e protetta da tutte le altre per cui, ad esempio, due programmi uguali che, in due sessioni distinte, attribuiscano valori diversi agli stessi parametri del programma, continueranno a trovarseli diversi: non c'è, in definitiva, alcuna sovrapposizione.
La parte di ambiente che invece ci tocca più da vicino, è senz'altro quella legata alle "variabili di ambiente": una serie di valori, cioè, che rappresentano una caratterizzazione di un determinato ambiente operativo.
Abbiamo già incontrato i comandi che definiscono i valori delle variabili di ambiente principali (dette anche variabili di sistema): i nomi di queste variabili, che coincidono tra l'altro con i nomi dei comandi, sono:

PATH	per la definizione dei percorsi di ricerca comandi;
DPATH	per la definizione dei percorsi di ricerca di file non eseguibili (in modo reale non esiste un comando per la definizione di questa variabile, tuttavia il comando APPEND visto in precedenza, con l'opzione /E ha il medesimo effetto);
PROMPT	per la definizione del formato del prompt con cui il sistema operativo ci richiede i comandi.
COMSPEC	è il nome di un'altra variabile di sistema che non ha un comando con il medesimo nome per la sua definizione: definisce il pathname del file che contiene il command processor per l'ambiente corrente e il suo valore è solitamente stabilito tramite i comandi SHELL e PROTSHELL in CONFIG.SYS (capitolo 9).

Il comando SET permette di visualizzare, annullare e alterare i valori di tutte le variabili di ambiente, ivi comprese quelle di sistema: digitando semplicemente SET, i valori di tutte le variabili di ambiente attualmente dichiarate vengono visualizzati; digitando SET seguito solo dal nome variabile e dal segno "=", il contenuto di quella variabile viene cancellato e la variabile stessa viene eliminata dall'ambiente;

digitando infine SET nome variabile=valore la variabile viene creata se inesistente e le viene assegnato il valore specificato oppure, se esisteva in precedenza, le viene assegnato il nuovo valore specificato cancellando il precedente.
Occorre fare un minimo di attenzione sulla diversa sintassi utilizzata dal comando SET per il trattamento delle variabili di sistema rispetto ai comandi che operano direttamente su di esse. Ad esempio, il comando:

```
PATH ;
```

come abbiamo già visto in precedenza, annulla tutti gli eventuali PATH precedentemente memorizzati: tramite il comando SET, otteniamo lo stesso risultato digitando:

```
SET PATH=
```

come si deduce dalle regole precedentemente esposte. Se digitassimo:

```
SET PATH ;
```

riceveremmo un messaggio di errore in quanto la sintassi dei comandi SET e PATH è differente.
Oltre le variabili di sistema, utilizzate appunto dal sistema operativo, le variabili di ambiente definite tramite il comando SET possono essere utilizzate nei file "batch" (capitolo 7) e inoltre il loro valore è accessibile a tutti i programmi che possono utilizzarle, ad esempio, per localizzare una directory dove si trovano dati necessari al loro funzionamento. Ad esempio, digitando:

```
SET LIB=C:\LIB
```

creiamo una variabile di ambiente, LIB appunto, che viene usata da molti compilatori per la ricerca delle librerie necessarie alla compilazione dei programmi.

SETCOM40

Ambiente: DOS

Tipo: *Esterno*

Il comando SETCOM40 permette ai programmi in modo reale di utilizzare le porte seriali COM1, COM2 e COM3.

Sintassi SETCOM40 COMn={ON|OFF}

Questo comando permette ai programmi in emulazione DOS l'utilizzo diretto delle porte seriali COM1, COM2 e COM3 (quest'ultima solo per la classe di calcolatori Personal System/2). Poiché i programmi DOS non sono stati concepiti per l'utilizzo in multitask, essi accedono alla porta seriale senza preoccuparsi di verificare che altri programmi la stiano già utilizzando: per prevenire una sovrapposizione, ad esempio, di stampe, al momento dell'accensione OS/2 predispone le porte seriali affinché siano utilizzate esclusivamente da sessioni avviate in modo protetto. Il comando SETCOM40 va utilizzato in modo reale perché altrimenti i programmi in DOS che accedono alla porta seriale, non riuscendovi, visualizzerebbero un messaggio di errore. SETCOM40 con valore ON abilita la porta specificata per l'utilizzo in modo reale, con valore OFF la disabilita; n individua il numero della porta.
In altri termini, per le sessioni in modo protetto è OS/2 che si fà carico di evitare la sovrapposizione di accessi alla medesima porta seriale, mentre per la sessione in modo reale è a nostro carico il controllo sulla disponibilità di una porta seriale e la sua conseguente abilitazione tramite l'utilizzo del comando SETCOM40. Se, ad esempio, vogliamo abilitare la porta COM1 per utilizzare una stampante seriale in emulazione DOS, dobbiamo digitare:

```
SETCOM40 COM1=ON
```

È da notare che affinché questo comando ottenga l'effetto desiderato, occorre comunque aver specificato il comando DEVICE=COM01.SYS (o COM02.SYS) in CONFIG.SYS (capitolo 9).

SORT

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando SORT è un filtro che permette di visualizzare i dati che un generico programma presenta sul video in linee ordinate, secondo l'ordinamento dei caratteri previsto per quella nazione.

Sintassi SORT [/R][/+numero colonna]
(dati provenienti da STANDARD INPUT)

Il comando SORT permette di ordinare, secondo l'ordine alfabetico stabilito tramite il comando CODEPAGE (capitolo 9), le linee che un generico programma presenterebbe in ordine casuale. Le lettere maiuscole e minuscole vengono considerate equivalenti nell'ordinamento. Il significato delle opzioni è il seguente:

/R Inverte il senso dell'ordinamento, effettua cioè l'ordinamento considerando l'ordine alfabetico della nazione stabilita come invertito, con il primo carattere al posto dell'ultimo.

/+numero colonna Indica la posizione da cui, per ciascuna linea, si vuole iniziare l'ordinamento. Numero colonna è un numero intero che può assumere tutti i valori da 1 al numero massimo di caratteri di una generica linea da ordinare. Se questa opzione non è specificata, SORT assume questo valore uguale a 1, cioè effettua l'ordinamento su tutti i caratteri di cui ciascuna linea è composta.

Analogamente al comando MORE, visto più sopra, il comando SORT non ha argomenti, e riceve i dati che poi mostra tramite lo STANDARD INPUT, cioè il "canale" di comunicazione a cui è solitamente collegata la tastiera. Anche per il SORT è necessario far sì che i dati visualizzati da altri programmi vengano incanalati nello STANDARD INPUT: questo si ottiene con l'utilizzo del segno "|" quando i dati da ordinare tramite SORT provengono da un programma, e con l'utilizzo del segno "<" quando i dati da ordinare provengono da un file.
Questo comando è un filtro e verrà trattato nel capitolo 7, possiamo tuttavia fare un esempio su come ottenere la directory di un disco in ordine alfabetico:

```
DIR | SORT
```

o ordinata per lunghezza decrescente dei file (la dimensione dei file inizia a colonna 14, considerando le linee presentate dal comando DIR):

```
DIR | SORT /R /+14
```

Comandi correlati: COUNTRY e CODEPAGE in CONFIG.SYS

SPOOL

Ambiente: OS/2

Tipo: *Esterno*

l comando SPOOL intercetta i dati che diversi programmi indirizzano a una stampante e li accoda per evitare sovrapposizioni.

Sintassi SPOOL [pathname] [/D:{PRN|LPT1}|LPT2|LPT3] [/O:dispositivo di stampa]

Questo comando va specificato per ogni stampante collegata al sistema per cui si intenda gestire l'accodamento delle stampe. SPOOL permette di definire un dispositivo di stampa diverso da quello a cui sono diretti i dati provenienti da un generico programma: in altri termini SPOOL, permette di "reindirizzare" l'uscita che un programma ha predisposto su una delle stampanti parallele LPT1 (o l'equivalente PRN), LPT2 o LPT3 sia sulle stampanti parallele stesse che su stampanti seriali connesse alle porte COM1, COM2 o COM3. Le opzioni del comando SPOOL hanno il significato seguente:

pathname — Questo parametro specifica il nome della directory dove SPOOL memorizza i file temporanei che verranno successivamente mandati in stampa. Se specificata, la directory deve già esistere quando il comando SPOOL viene attivato. Se non viene specificato alcun valore, SPOOL utilizza la directory \SPOOL, predisposta automaticamente durante la fase di installazione di OS/2.

/D — Questo parametro, che può assumere solo uno dei valori che individuano le stampanti parallele, stabilisce a quale stampante vogliamo che SPOOL intercetti le stampe per accodarle poi verso la stampante specificata dall'opzione seguente. Il valore di default è LPT1.

/O:dispositivo di stampa — Questo parametro individua il dispositivo su cui fisicamente avverrà la stampa:
può assumere i valori LPT1 (o l'equivalente PRN), LPT2, LPT3, COM1, COM2 e, solo per la classe di computer Personal System/2, COM3. Il valore assunto per default è quello specificato dall'opzione /D.

Come si vede dalla sintassi del comando, affinché i programmi possano utilizzare SPOOL occorre che le stampe da essi prodotte siano indirizzate verso le stampanti parallele.
Non è necessario avviare manualmente l'esecuzione del comando SPOOL in

quanto esso non è un programma interattivo: il metodo più efficiente per utilizzarlo è includere in CONFIG.SYS (capitolo 9) il comando RUN che attivi lo SPOOL per una determinata stampante. Altra possibilità è avviare il comando SPOOL tramite il comando DETACH visto più sopra. Se, ad esempio, al nostro sistema è collegata una stampate parallela come LPT1, occorrerà includere in CONFIG.SYS questo comando:

```
RUN=\OS2\SPOOL
```

se inoltre è collegata una stampante seriale, supponendo di utilizzarla con il nome della stampante LPT2, occorrerà includere questo comando in CONFIG.SYS:

```
RUN=\OS2\SPOOL /D:LPT2 /O:COM1
```

In entrambi i casi la directory dove SPOOL memorizzerà i file temporanei è quella di default, \SPOOL.

Occorre fare alcune considerazioni riguardo alle stampanti seriali: innanzitutto occorre che il comando DEVICE=COM01.SYS (o COM02.SYS) sia stato incluso nel file CONFIG.SYS (capitolo 9) affinché il sistema sia in grado di utilizzare le porte seriali. Nell'esempio precedente, inoltre, la stampante dev'essere selezionata per i parametri di comunicazione di default (cioè 1200 baud, 7 databits, parità pari e 1 stopbit, come specificato nella trattazione del comando MODE, più sopra): se non è così, occorre eseguire il comando MODE per predisporre i valori corretti. Poiché non è possibile eseguire alcun comando durante la fase di avviamento del sistema, fase in cui vengono eseguiti i comandi contenuti in CONFIG.SYS, non è possibile avviare SPOOL per una stampante seriale che abbia dei parametri di comunicazione diversi da quelli di default, tramite il comando RUN in CONFIG.SYS. Per attivare lo SPOOL per una stampante con queste caratteristiche occorre comunque che in CONFIG.SYS sia stato specificato il comando

```
DEVICE=COM01.SYS (o COM02.SYS),
```

dopodiché occorre, al prompt del sistema operativo, selezionare tramite il comando MODE i parametri di comunicazione adatti per detta stampante; infine, sempre dal prompt del sistema operativo, possiamo avviare il comando SPOOL tramite il comando DETACH. Supponendo di voler attivare SPOOL per la stampante dell'ultimo esempio, selezionata però a 9600 baud, 8 databits, nessun controllo di parità e con controllo del flusso di trasmissione, dovremo utilizzare i seguenti comandi:

```
MODE COM1:96,N,8,XON=ON
DETACH SPOOL /D:LPT2 /O:COM1
```

Come vedremo nel capitolo 7, questi due comandi possono essere inclusi in un file

"batch" affinché vengano eseguiti automaticamente all'avviamento del sistema.
Il comando PRINT, visto più sopra, interagisce con il comando SPOOL quando utilizziamo le opzioni /C e /T del comando PRINT stesso.
Il comando SPOOL inizia effettivamente a stampare i dati provenienti da un programma e memorizzati in un file temporaneo nella directory predisposta, quando il programma stesso dà il via. I programmi eseguiti in emulazione DOS non hanno quest'ultima caratteristica, ed è pertanto a nostro carico iniziare la stampa premendo contemporaneamente i tasti Ctrl, Alt e PrtSc perché, altrimenti, la stampa inizierebbe solo al termine del programma.

Comandi correlati: PRINT, MODE

START

Ambiente: OS/2

Tipo: *Interno*

Il comando START permette di attivare un programma e un nuovo screen group.

Sintassi START [titolo] [/C] [comando] [argomento] [...]

Questo comando permette di attivare un programma dalla sessione corrente, ponendolo però in una nuova sessione: in altri termini, l'utilizzo del comando START equivale alla attivazione di uno dei programmi presenti nella colonna "Start a program" del program selector (capitolo 4).
Titolo è il nome che si vuole compaia nel program selector e può avere una lunghezza massima di 30 caratteri: va racchiuso tra una coppia di virgolette (" ") e, se la lunghezza eccede la massima permessa, i caratteri eccedenti vengono trascurati. Se non è specificato, nel program selector comparirà il nome del comando specificato.
L'opzione /C lega la durata della sessione alla durata del comando: al termine dell'esecuzione del comando la sessione verrà automaticamente eliminata dal program selector. Inoltre, se questa sessione è presente sul monitor quando il programma termina, il suo posto viene preso dal program selector.
Gli argomenti che opzionalmente seguono il comando di cui si richiede l'esecuzione sono quelli eventualmente richiesti da quest'ultimo, in quanto gli unici argomenti riconosciuti dal comando START sono quelli che precedono il nome del comando da eseguire.
L'esecuzione del comando START, senza alcun argomento, inizia una nuova sessione in modo protetto senza eseguire alcun comando, equivale cioè alla selezione di "OS/2 Command Prompt" dal program selector.

Comandi correlati: CMD

SUBST

Ambiente: DOS

Tipo: *Esterno*

Il comando SUBST permette di identificare una directory con il nome di un drive.

Sintassi SUBST [drive: drive:\directory|drive: /D]

L'assegnazione del nome di un drive, inesistente come unità fisica, a una directory di un altro drive permette di accedere a detta directory utilizzando l'identificatore del drive. SUBST, seguito dall'opzione /D, pone fine all'assegnazione relativa a drive; senza alcun argomento mostra le assegnazioni in essere.
Se, ad esempio, il nostro drive corrente è C e noi vogliamo accedere alla directory CONDOMIN\VIA_ROMA\BIANCHI come fosse il drive E, che non esiste nel nostro calcolatore, digiteremo:

```
SUBST E: C:\CONDOMIN\VIA_ROMA\BIANCHI
```

dopodiché digitare E: sarà equivalente all'esecuzione del comando CD\CONDOMIN\VIA_ROMA\BIANCHI sul nostro drive corrente. In effetti, il comando DIR, eseguito dopo il precedente, mostrerebbe lo stesso elenco di file che vedremmo digitando DIR dopo avere digitato E:. Digitando:

```
SUBST E: /D
```

cancelliamo l'assegnazione precedente cosicché ogni tentativo di effettuare operazioni sul drive E provocherà la visualizzazione di un messaggio di errore inerente l'inesistenza del drive suddetto.

Comandi correlati: JOIN, ASSIGN

SYS

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando SYS copia i file nascosti contenenti il sistema operativo.

Sintassi SYS drive:

Questo comando permette di trasferire i due file che contengono il sistema operativo. Nella versione Microsoft, i due file si chiamano OS2BIO.COM e OS2DOS.COM, mentre nella versione IBM si chiamano IBMBIO.COM e IBMDOS.COM. Questi file sono nascosti e, infatti, non è possibile visualizzarli tramite il comando DIR. L'unica prova della loro esistenza è il numero dei file nascosti riportato dal comando CHKDSK. Il comando SYS permette di trasferire questi due file sul drive indicato come destinazione prelevandoli dal disco corrente; pertanto occorre che, al momento della digitazione del comando, il drive corrente sia quello da cui avviene l'avviamento del sistema (quindi A o C) affinché contenga i file suddetti.
Il disco di destinazione di questi due file deve essere stato formattato o, in alternativa, doveva già contenere i file di una precedente versione del sistema operativo, sia DOS che OS/2. Questo perché questi due file devono occupare una posizione ben precisa sul disco, pertanto non possono venir copiati su un disco già utilizzato per altri scopi, in quanto la posizione suddetta risulterà presumibilmente occupata da altri file.

Comandi correlati: FORMAT

TIME

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando TIME permette di visualizzare e di modificare l'orario corrente.

Sintassi TIME [ora]

Il comando TIME digitato senza alcun argomento mostra l'orario, analogamente al comando DATE visto più sopra, nel formato specificato per il paese selezionato dal comando COUNTRY in CONFIG.SYS (capitolo 9) e si predispone ad accettarne la variazione nel medesimo formato.
L'orologio interno al calcolatore è un orologio con un ciclo di 24 ore, e le regole per l'immissione dell'ora sono le seguenti:

- l'ora va indicata con un numero compreso fra 0 e 24;
- i minuti primi vanno indicati con un numero compreso fra 0 e 59, se omessi, viene assunto il valore 0;
- i minuti secondi vanno indicati con un numero compreso fra 0 e 59, se omessi, viene assunto il valore 0;
- i centesimi di secondo vanno indicati con un numero compreso fra 0 e 99, se omessi, viene assunto il valore 0.

Se alle 15 e 30 digitiamo TIME compare il seguente messaggio:

```
The current time is: 15:30:19,24
Enter the new time (dd-mm-yy):
```

cui possiamo rispondere digitando un nuovo orario o premendo semplicemente ENTER per confermare quello inserito. Il comando TIME può anche essere seguito direttamente dal nuovo orario da impostare: in questo caso non comparirà alcuna richiesta, salvo che non si commetta un errore nell'impostarlo.

Comandi correlati: comando COUNTRY in CONFIG.SYS, DATE, PROMPT

TRACE

Ambiente: OS/2

Tipo: *Esterno*

TRACE è un comando di manutenzione che abilita o disabilita la memorizzazione degli eventi relativi al sistema operativo.

Sintassi TRACE {ON|OFF} [codice evento][,codice evento][...]

Lo scopo principale di questo comando è permettere l'osservazione degli eventi operativi in cui è coinvolto il sistema operativo, questo per verificarne il comportamento in rapporto all'utilizzo con applicativi complessi.
L'abilitazione o meno del TRACE può essere effettuata dall'omonimo comando TRACE in CONFIG.SYS (capitolo 9): se quest'ultimo non è specificato, il valore di default di TRACE è OFF.
Il codice evento si riferisce ad una codifica delle classi di eventi che sono osservabili con il comando TRACE: non indicando nessun codice, si sottintende che l'opzione scelta, sia ON che OFF, ha effetto su tutti i codici, altrimenti viene attivata (ON), o disattivata (OFF), l'osservazione dei soli eventi il cui codice è elencato di seguito all'opzione scelta.
Diamo di seguito un elenco dei codici evento più importanti:

CODICE EVENTO	TIPO EVENTO
00	Tutte le classi (equivale a non specificare alcun codice)
01	Riservato
02	Transizioni da modo reale a modo protetto e viceversa
03	"Exceptions" del microprocessore
04	"Interrupts" del microprocessore
05 - 15	Riservati

Esiste un significato anche per i codici fino a 255, si tratta comunque di un livello di approfondimento non previsto per questo testo.
Al fine di permettere la memorizzazione degli eventi selezionati con l'opzione ON di questo comando, occorre che sia stato riservato un apposito spazio nella memoria RAM tramite il comando TRACEBUF in CONFIG.SYS (capitolo 9).

Comandi correlati: TRACE e TRACEBUF in CONFIG.SYS, TRACEFMT

TRACEFMT

Ambiente: OS/2

Tipo: *Esterno*

Il comando TRACEFMT permette di visualizzare gli eventi che sono stati memorizzati con il comando TRACE.

Sintassi TRACEFMT

L'utilizzo di questo comando è chiaramente limitato al caso in cui si stia eseguendo, o sia stato eseguito in precedenza, il TRACE (sia tramite il comando precedente che perché attivato in CONFIG.SYS) degli eventi relativi al sistema operativo.
Quanto visualizzato da TRACEFMT va normalmente sottoposto ad un esperto di sistemi operativi affinché possa individuare eventuali problemi.
Non c'è nessun limite alla ripetizione del comando TRACEFMT: a differenza del comando CREATEDD visto più sopra, non ha alcun effetto sulla funzionalità del sistema operativo. In effetti CREATEDD interviene su tutta la memoria RAM in cui è localizzato il sistema operativo, mentre TRACEFMT riporta il contenuto di un'apposita area di memoria predisposta dal comando TRACEBUF in CONFIG.SYS.

Comandi correlati: TRACE e TRACEBUF in CONFIG.SYS, TRACE

TREE

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando TREE visualizza la struttura del file system del drive specificato.

Sintassi TREE [drive:] [/F]

Il comando TREE mostra sul video il nome di tutte le directory del drive specificato, di quello corrente se non ne specifichiamo nessuno, dando per ciascuna l'elenco delle directory contenute, ripetendo poi il ciclo per il livello sottostante e così via per tutte le directory del file system.
Normalmente il comando TREE visualizza solo il pathname completo di ogni directory e il nome delle directory contenute, tuttavia tramite l'opzione /F è possibile avere anche, per ciascuna directory, l'elenco di tutti i file contenuti.
Se, ad esempio, volessimo vedere tutte le directory del drive C, che è il nostro drive corrente, dovremmo digitare semplicemente:

```
TREE
```

e otterremmo il risultato desiderato.

Comandi correlati: DIR, CHDIR, MKDIR, RMDIR

TYPE

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando TYPE permette di visualizzare sullo schermo il contenuto di file di testo.

Sintassi TYPE pathname [pathname] [...]

Il pathname deve individuare un file di testo, cioè scritto tramite un editor come, ad esempio, EDLIN, che vedremo nel prossimo capitolo.
In emulazione DOS è permesso fornire un solo pathname al comando, pathname che inoltre non può contenere i caratteri "?" e "*". In modo protetto, possono essere forniti più pathname al comando che possono quindi anche contenere i caratteri suddetti.
Se il comando TYPE viene richiamato con il pathname di un file non di testo, come ad esempio il comando COMMAND.COM, sul monitor compaiono una serie di caratteri indecifrabili e durante la visualizzazione viene emesso più volte il segnale di "allarme" (il classico "beep") da parte del calcolatore. Normalmente ciò non causa alcun danno, tuttavia è bene evitare di utilizzare il comando TYPE su file di cui non si ha la certezza che siano file di testo. Del resto, non esiste alcun modo per sapere a priori se un file è o meno di testo: l'unica prassi ormai consolidata è di considerare di testo i file che hanno estensione TXT, BAT, CMD, LST, C, COB, CBL, MAC, ASM, PAS e qualche altro.
Se, ad esempio, volessimo visualizzare il contenuto del file STARTUP.CMD che si trova nella directory root del drive C, dovremmo digitare:

```
TYPE C:\STARTUP.CMD
```

In questo modo il comando è utilizzabile sia in modo reale che protetto; digitando invece:

```
TYPE C:\*.CMD
```

otterremmo la visualizzazione del contenuto di tutti i file presenti nella directory root del drive C con estensione CMD, ma solo in modo protetto in quanto in modo reale questo comando darebbe origine semplicemente ad un messaggio di errore.

Comandi correlati: MORE

VER

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando VER visualizza il numero della versione corrente di OS/2.

Sintassi VER

Questo comando visualizza il numero della versione corrente del sistema operativo. Sia per la versione distribuita dalla IBM che per quella distribuita dalla Microsoft, la versione a cui si riferisce questo volume è la 1.00, cioè la prima versione disponibile di OS/2.

Comandi correlati: PROMPT

VERIFY

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando VERIFY abilita o disabilita la verifica della corretta memorizzazione di dati sul disco.

Sintassi VERIFY [ON|OFF]

Il comando VERIFY ha il medesimo effetto dell'opzione /V vista per il comando COPY più sopra e per il comando XCOPY più oltre. Lo scopo di questo comando è far sì che ogni volta che alcuni dati vengono memorizzati su un supporto di memoria di massa, questi dati vengano immediatamente riletti in modo da garantire una loro futura reperibilità.
L'opzione ON attiva questa funzione, mentre l'opzione OFF la disattiva (questo è anche il valore di default). Il comando VERIFY senza alcuna opzione mostra lo stato di questa funzione.

Comandi correlati: COPY, XCOPY

VOL

Ambiente: *Entrambi*

Tipo: *Interno*

Il comando VOL visualizza il nome assegnato ad uno o più dischi.

Sintassi VOL [drive:] [...]

Assegnare un nome a un disco è un'operazione opzionale che viene effettuata o in fase di formattazione, specificando l'opzione /V al comando FORMAT, o successivamente con il comando LABEL, entrambi visti più sopra.
Questo comando permette di visualizzare, se esiste, il nome dato alle unità specificate o a quella corrente, con l'avvertenza che in modo reale è possibile richiedere il nome di un drive per volta, mentre in modo protetto possono essere forniti più argomenti.

Comandi correlati: FORMAT, LABEL

XCOPY

Ambiente: *Entrambi*

Tipo: *Esterno*

Il comando XCOPY permette di effettuare delle copie di file e directory.

Sintassi XCOPY pathname [pathname] [/D:data] [/S] [/E] [/P] [/V] [/A] [/M]

Questo comando permette la copia di intere parti del file system. A differenza del comando COPY, visto più sopra, che può copiare un file, più file e al limite tutti i file di una directory (escluse però le eventuali sottodirectory), XCOPY può copiare, data una directory, tutti i file in essa contenuti, le directory in essa contenute con i rispettivi file e, se vi fossero altre directory contenute in queste ultime XCOPY le copierebbe, con tutti i loro file e così di seguito.
XCOPY è, ad esempio, molto comodo se si vuole copiare un intero disco su di un altro di formato diverso (tale cioè da non permettere l'utilizzo del comando DISKCOPY), quando le ramificazioni del file system renderebbero la realizzazione di tale copia con il comando COPY una vera e propria punizione.
Il primo pathname individua il/i file o la directory che si vuole copiare. Se è specificato un secondo pathname, questo può essere, in dipendenza anche dal primo, un nome di file o directory: se non è specificato alcun pathname, la copia avverrà a partire dalla nostra directory corrente. Il significato delle opzioni è il seguente:

/D:data Copia soltanto i file specificati la cui data di creazione o di ultima modifica è uguale o maggiore di data. La data, analogamente ai comandi BACKUP e RESTORE, va specificata nel formato stabilito dal comando COUNTRY (nazione, capitolo 9) che, nel caso dell'Italia, è GG/MM/AA (cioè due cifre per il giorno, 2 per il mese e le ultime 2 dell'anno).

/S Questa opzione provoca la copia di tutte le directory, con i rispettivi file, contenute nella directory definita dal primo pathname e di quelle a loro volta contenute nelle precedenti e così via fino all'ultimo livello. Se una directory è vuota non viene copiata; cioè se non esiste nel path di destinazione non viene creata e se esiste non viene in alcun modo alterata.

/E Questa opzione è efficace solo se utilizzata insieme alla precedente in quanto provoca la creazione delle directory che, vuote nel path sorgente, non esistono nel path di destinazione.

/P Con questa opzione, XCOPY chiede conferma per ogni file che si appresta a copiare: rispondendo Y il file viene copiato,

rispondendo N il file non viene copiato; in ogni caso XCOPY passa ad esaminare il file successivo riprendendo il ciclo. Tramite questa opzione è possibile effettuare una copia selettiva.

/V Analogamente al comando COPY, questa opzione obbliga il sistema operativo ad effettuare la verifica di tutti i dati che vengono copiati per prevenire errori di scrittura su disco che precluderebbero il futuro utilizzo dei dati copiati.

/A La presenza di questa opzione provoca la copia dei soli file che hanno l'attributo archivio attivo (vedi comando ATTRIB), senza peraltro alterarlo.

/M Questa opzione agisce come la precedente /A, con la sola differenza che l'attributo archivio dei file copiati viene disattivato, permettendo così una copia controllata, analogamente al comando BACKUP.

Le opzioni /A e /M vengono considerate nell'ordine di digitazione per cui, sesono stati specificate entrambe, verrà presa in considerazione solo la seconda.
Se, ad esempio, vogliamo copiare la struttura utilizzata nell'esempio del capitolo 3, per quanto riguarda la directory CONDOMIN, sul disco posto nel drive A, digiteremo:

```
XCOPY C:\CONDOMIN A: /S
```

tenendo presente che sul disco A non verrà creata la directory CONDOMIN, ma i file e le directory saranno posizionati a partire dalla directory root del disco A. Se avessimo voluto realizzare esattamente la medesima struttura di C:\CONDOMIN in A, avremmo dovuto digitare prima:

```
MKDIR A:\CONDOMIN
```

e, successivamente:

```
XCOPY C:\CONDOMIN A:\CONDOMIN /S
```

dopodiché la struttura presente sul drive A sarebbe stata la replica di quella creata sul drive C.

Comandi correlati: BACKUP, COPY, VERIFY

Prima di concludere il capitolo sui comandi, elenchiamo tutti quelli che non possono essere utilizzati in rete (per rete di calcolatori si intende il collegamento di più calcolatori più il software necessario per sfruttare da ogni calcolatore le

risorse di tutte gli altri). Un comando non è utilizzabile in rete quando la sua funzionalità è limitata al calcolatore su cui si trova; mentre può operare in rete se la stessa operazione che compie sui dispositivi connessi al calcolatore "locale" (cioè quello in cui si trova il comando) può essere eseguita su tutti gli altri calcolatori connessi in rete (calcolatori "remoti"). I seguenti comandi possono essere utilizzati solo localmente:

CHKDSK
DISKCOMP
DISKCOPY
FDISK
FORMAT
LABEL
RECOVER
SUBST
SYS

Questi stessi comandi, inoltre, presentano un messaggio di errore se vengono eseguiti su drive che siano stati oggetto di reindirizzamento da parte dei comandi JOIN e SUBST (escluso quindi quest'ultimo). Inoltre, sempre gli stessi comandi, ignorano ogni assegnazione fatta tramite il comando ASSIGN.

CAPITOLO SEI

EDLIN: un programma per gestire file di testo

CARATTERISTICHE GENERALI DELL'EDLIN

L'Edlin è un programma che permette la redazione testi. È un prodotto molto semplice in quanto agisce su una sola linea per volta, contrariamente ad altri programmi di trattamento testi, comunemente denominati *full screen editor* nei quali è possibile agire indifferentemente su qualsiasi riga del testo (in anglosassone *full screen* significa "a tutto schermo"). Questo programma è adatto soprattutto al trattamento di testi non eccessivamente lunghi, in caso contrario la redazione di un documento diventa molto laboriosa.
La versione di Edlin fornita con OS/2 è uguale a quella fornita con la versione 3 del DOS.
Possiamo utilizzare il programma EDLIN solo in modo reale cioè quando il nostro calcolatore emula l'ambiente DOS.
Il testo costruito con il programma EDLIN può essere composto da varie righe, nessuna delle quali può però superare la lunghezza massima di 253 caratteri.
Ogni riga o linea del testo viene numerata in modo automatico dal programma EDLIN: questo ci consente di sapere quale riga del testo stiamo trattando e consente ad EDLIN di sapere quale linea deve rendere disponibile al trattamento. La numerazione delle linee viene però eliminata al momento della memorizzazione del file su disco non essendo essa una parte del testo, ma un utile riferimento per il trattamento del testo stesso. Ogni volta che cancelliamo una riga dal testo le linee rimaste vengono rinumerate, decrementando il numero dalla riga successiva a quella eliminata. Analogamente, se inseriamo una nuova riga, le righe successive a quella inserita vengono incrementate.

COME ATTIVARE EDLIN

Cominciamo con ordine e vediamo innanzitutto come far partire il programma EDLIN. La sintassi è:

```
EDLIN pathname
```

Supponiamo di voler scrivere un file di testo che intendiamo chiamare "testo.txt". Supponendo di trovarci già nella directory in cui vogliamo creare detto file, digiteremo il comando:

```
EDLIN TESTO.TXT
```

A questo punto il sistema operativo trasferisce in memoria il programma EDLIN e lo avvia. EDLIN verifica l'esistenza del file da noi indicato, e, se non esiste, presenta il seguente messaggio seguito immediatamente dal prompt dell'editor:

```
EDLIN has created a new file
*
```

Se EDLIN avesse verificato l'esistenza di un file con il nome da noi digitato, il messaggio presentato da EDLIN sarebbe stato:

```
The entire input file has been read
*
```

L'asterisco ci indica che l'editor è in attesa di un comando ed è il prompt del programma EDLIN. EDLIN non opera direttamente sul nome proposto, crea invece un file temporaneo, su cui effettuare tutte le modifiche, con il medesimo nome da noi indicato ma con estensione $$$, tipica dei file temporanei. Il nome effettivo con l'estensione da noi indicata viene attribuito al momento del salvataggio del nostro file su disco.
Se supponiamo che il file "testo.txt" sia stato creato da EDLIN, l'unico comando che siamo in grado di attivare in questo momento, trattandosi appunto di un nuovo file, è il comando I (sigla di "insert", cioè inserimento). In risposta all'attivazione di questa funzione EDLIN ci mostra il numero assegnato alla riga (nel nostro caso 1) seguito da un asterisco che serve a segnalarci che quella è la riga che stiamo trattando. A questo punto possiamo digitare il nostro testo: la funzione inserimento, una volta attivata, termina solo con la pressione simultanea dei tasti CTRL e BREAK.
Se, invece, abbiamo digitato il nome di un file esistente, EDLIN carica in memoria tante linee quante l'ammontare massimo della memoria ad esso dedicata, cioè circa il 75% del totale disponibile, ne può contenere. Il prompt dell'editor ci mette questa volta nella condizione di trattare il file (o parte di esso) che è stato trasferito

in memoria.
Se la lunghezza del file eccedeva la memoria disponibile, per gestire la restante parte del nostro testo dobbiamo scrivere su disco, servendoci del comando W (sigla di "write", cioè scrittura), illustrato più in dettaglio nella parte riservata ai comandi, una parte delle righe già modificate e caricare in memoria il resto del nostro testo servendoci del comando A (sigla di "append", cioè accodamento).
Sia nella redazione di un nuovo testo che nel trattamento di un testo già esistente, per terminare l'uso del programma EDLIN possiamo utilizzare due comandi, Q (sigla di "quit", cioè abbandono) ed E (sigla di "end", cioè fine). Come tutti gli altri, anche questi comandi devono essere collocati all'inizio della linea, subito dopo il carattere * che come abbiamo già visto rappresenta il prompt dell'editor.
Q ci consente di abbandonare l'editor senza che le eventuali modifiche da noi eseguite aggiornino il file indicato; E, al contrario, riporta sul file indicato tutte le modifiche effettuate. In entrambi i casi il file temporaneo, contraddistinto dall'estensione .$$$, viene automaticamente eliminato e riappare il prompt del sistema operativo.

Attenzione: Non bisogna utilizzare nei nomi che diamo ai nostri file l'estensione .BAK, in quanto questa estensione è riservata all'editor per la copia, che esso effettua automaticamente, del file che stiamo trattando. Il file di copia .BAK, creato quando operiamo su file preesistenti sul disco, permette di conservare la versione precedente del nostro testo nel caso che le modifiche effettuate non siano più di nostro gradimento.

TRATTAMENTO DEI FILE BINARI

Se nel file che vogliamo trattare si trova, frammisto ai normali caratteri, il carattere detto "Control Z" (perché ottenuto dalla pressione simultanea dei tasti Ctrl e Z e utilizzato per molto tempo come indicatore di fine del file), quando richiamiamo l'EDLIN con il pathname di questo file, esso interpreterà questo carattere come indicatore della fine del file, interrompendone il caricamento anche se il file non è ancora terminato. Possiamo ovviare a questo inconveniente tramite un'opzione del programma EDLIN:

```
EDLIN TESTO.TXT /B
```

L'opzione /B evita che EDLIN interpreti un carattere di fine file, eventualmente contenuto in un testo, obbligandolo a trattare detto file rispetto alla lunghezza totale e non al contenuto.
Se vogliamo invece immettere il carattere suddetto in un qualsiasi punto del nostro testo, dobbiamo attivare il comando di inserimento, quindi premere contemporaneamente i tasti Ctrl e V, seguiti dalla lettera Z. La combinazione dei caratteri Ctrl e V indica all'editor che il carattere di immissione successivo è un carattere di

controllo e la lettera maiuscola Z viene quindi trasformata nel carattere di controllo Ctrl Z. La combinazione Ctrl e V può essere seguita da qualsiasi carattere onde ottenere tutti i possibili caratteri di controllo.

I COMANDI DI EDLIN

Prima di descrivere i comandi del programma EDLIN, di seguito elencati in ordine alfabetico, diamo delle indicazioni generali applicabili a tutti i comandi:

- il prompt è un asterisco;
- esiste sempre una linea corrente, o di riferimento, cui si riferiscono i parametri di spostamento relativo: è caratterizzata da un asterisco posto fra il numero della linea e il suo contenuto;
- ogni comando viene attivato digitando la lettera che lo contraddistingue, fatta eccezione per il comando edit;
- possiamo far precedere o seguire ogni comando da parametri. Gli unici comandi cui non necessitano parametri sono E (end) e Q (quit);
- possiamo scrivere i comandi indifferentemente in maiuscolo o minuscolo;
- non occorre immettere nessun elemento separatore fra i comandi e i parametri. È obbligatorio, invece immettere come separatori o una virgola o uno spazio quando facciamo riferimento a due o più parametri;
- i comandi che immettiamo vengono interpretati solo dopo che abbiamo premuto il tasto ENTER;
- l'uscita da un comando si ottiene premendo i tasti Ctrl e C (o Ctrl e Break);
- se stiamo utilizzando un comando che ci mostra a video più righe di quelle che il nostro schermo è in grado di contenere, possiamo fermarne la visualizzazione con il tasto PAUSE e farla ripartire premendo qualsiasi tasto;
- possiamo fare riferimento a un numero di linea in maniera assoluta, utilizzando cioè il suo numero di linea, o in maniera relativa attraverso la simbologia + o –. Se, ad esempio, abbiamo modificato la riga del nostro testo identificata dal numero 10 e introduciamo il comando +3L l'editor ci visualizza la linea 10 e le tre righe seguenti. Quindi, relativamente all'ultima riga da noi trattata, possiamo riferirci alle righe immediatamente precedenti ad essa (con il simbolo –) o immediatamente successive ad essa (con il simbolo +). Se scriviamo –10, +10L otteniamo la visualizzazione di ventuno righe: le dieci righe precedenti, l'ultima riga trattata e le dieci righe successive alla nostra linea di riferimento;
- nella maggior parte dei comandi, possiamo sostituire un numero di linea con due simboli particolari. Il simbolo "#" corrisponde al primo numero di linea disponibile, il simbolo "." identifica la linea in corso;
- in linea di massima, i comandi vanno eseguiti uno di seguito all'altro. Possiamo tuttavia scrivere più comandi consecutivi in corrispondenza della stessa linea di comando.

COMANDO "A" ("APPEND", ACCODAMENTO)

Aggiunge al testo che stiamo trattando le linee non caricate dall'editor stesso per esaurimento della memoria disponibile.
Possiamo utilizzare questo comando solo per quei file che, a causa delle loro dimensioni, non possono essere caricati completamente in memoria. Se indichiamo il comando senza specificare il numero delle linee che desideriamo caricare, l'editor leggerà tante linee quante la memoria ne può contenere, fino al raggiungimento del 75% della memoria massima disponibile in ambiente DOS.
Naturalmente, affinchè il comando abbia effetto, prima di attivarlo dobbiamo scrivere su disco, servendoci del comando write, parte del nostro testo, in modo da liberare memoria sufficiente al caricamento delle restanti linee del nostro file.

COMANDI "C" ("COPY", COPIA)

Il comando C ci permette di effettuare la copia di una linea o di un gruppo consecutivo di linee in un punto determinato del nostro testo. Le linee che intendiamo copiare verranno posizionate a partire dal numero di linea da noi indicata come parametro di destinazione.La sintassi di questo comando è:

```
linea inizio,linea fine,linea destinazione,numero copieC
```

Il parametro linea inizio ci indica qual è la linea del nostro testo da cui intendiamo iniziare la copia. Il parametro linea fine ci indica l'ultima linea che intendiamo copiare. Se uno di questi due parametri viene omesso, l'editor attribuisce come linea inizio o linea fine il numero corrispondente alla linea in corso di trattamento. È invece obbligatorio il parametro linea destinazione per determinare a quale altezza del nostro testo deve avvenire l'inserimento delle linee da copiare. L'editor inserisce quindi le linee copiate a partire dalla linea da noi indicata come linea di destinazione della copia. Attraverso il parametro non obbligatorio numero copie indichiamo all'editor quante copie desideriamo effettuare delle linee selezionate. Se omettiamo l'indicazione del numero di copie, l'editor ne effettua una sola. In caso contrario, le copie si inseriranno una di seguito all'altra, a partire dalla linea di destinazione indicata. Volendo, ad esempio, copiare le linee dalla 1 alla 5, portarle alla 8 facendone 10 copie, scriveremo:

```
1,5,8,10C
```

Il comando ci esegue per dieci volte la copia delle linee comprese fra i numeri 1 e 5 posizionandoli alla linea contraddistinta dal numero 8. Il comando C non può essere utilizzato per modificare l'ordine delle linee del file. Non possiamo, quindi, utilizzare come numero di linea di destinazione un numero compreso, nel nostro esempio, fra 1 e 5 perché otterremmo da parte dell'editor la visualizzazione di un messaggio di errore.
Al termine di esecuzione del comando la prima linea copiata diventa la nostra attuale linea di trattamento.

COMANDO "D" ("DELETE", CANCELLAZIONE)

Questo comando ci permette di eliminare una o più righe di testo. Dopo l'eliminazione, l'editor effettua un immediato aggiornamento della numerazione. La sintassi di questo comando è:

```
linea inizio, linea fineD
```

dove con linea inizio indichiamo la prima riga che intendiamo cancellare e con linea fine indichiamo l'ultima riga da eliminare. Come per il comando di copia l'indicazione della linea di partenza e di arrivo non è obbligatoria. Se omettiamo, quindi, la specifica di questi due parametri, l'editor eliminerà la linea in corso di trattamento, posizionando il prompt nella linea immediatamente seguente quella eliminata. Volendo, ad esempio, eliminare dal nostro testo le righe che vanno dalla 1 alla 10, scriveremo:

```
1,10D
```

A seguito di questo comando otterremo la cancellazione delle linee dalla 1 alla 10 e la linea numero 11 verrà rinumerata dall'editor diventando nel nostro testo la linea numero 1. La nostra linea di trattamento è diventata la numero 1. Se ora proviamo ad immettere il comando:

```
,3D
```

l'editor eliminerà le righe che vanno da quella in corso di trattamento fino alla linea numero 3. Quindi nel caso del nostro esempio, con questo comando cancelliamo le linee che vanno dalla numero 1 alla numero 3.
Se immettiamo il comando:

```
5D
```

dal nostro testo viene eliminata solo la riga contrassegnata dal numero 5.

COMANDO "EDIT" (MODIFICA DELLA RIGA)

Questo comando permette di variare una riga del nostro testo. La sintassi del comando è molto semplice, basta infatti immettere il numero della riga che intendiamo variare.
Vediamo insieme un esempio per meglio chiarire il funzionamento di questo comando. Dopo aver visualizzato sul video le righe formanti il nostro documento:

```
    1:*   Linea di testo numero uno
    2:    Linea di testo numero due
    3:    Linea di testo numero tre
    4:    Linea di testo numero quattro
*
```

decidiamo di variare la riga 1: digitando 1 seguito da ENTER l'editor ci rende disponibile per la variazione la suddetta linea. Se avessimo premuto solo ENTER, l'editor ci avrebbe reso disponibile per la variazione la linea contrassegnata dal numero due. Questo perché l'editor in mancanza di altre indicazioni assume come presupposto che la riga che si intende variare è quella successiva a quella corrente.
Supponiamo di aver premuto ENTER: variamo quindi la linea due che si presenta sul nostro schermo:

```
    2:*   Linea di testo numero due
    2:*
```

Se premiamo il tasto ENTER a questo livello la linea resta invariata. Ma se abbiamo digitato anche un solo carattere, premendo ENTER confermiamo tutte le variazioni effettuate sulla riga, cancellandone così il contenuto dalla posizione attuale fino alla fine della linea.
Supponiamo di voler sostituire nr a numero: possiamo avanzare con la freccia a destra fino a posizionarci sotto la parola da variare, oppure premere il tasto funzione F2 e la lettera "u", e l'editor ci posizionerà sotto tale lettera riportando a sinistra la parte invariata:

```
    2:*   Linea di testo numero due
    2:*   Linea di testo n
```

A questo punto possiamo premere ENTER per confermare le variazioni, o premere il tasto ESC o i tasti Ctrl e Break per annullare le variazioni mantenendo inalterata la riga.

COMANDO "E" ("END", FINE)

Con questo comando termina l'utilizzo dell'editor e il nostro documento viene salvato su disco.
Se il file esisteva già, ne viene conservata la versione precedente in un file con il nome da noi attribuito più l'estensione .BAK. Se la dimensione del file modificato eccede lo spazio disponibile sul disco, la parte di documento che l'editor non è riuscito ad aggiornare viene persa. Ad indicare la mancata scrittura di tutto il documento sul nostro disco rimane il file temporaneo che ha estensione .$$$.

COMANDO "I" ("INSERT", INSERIMENTO)

Questo comando ci permette inserire nuove linee nel documento che stiamo trattando. È il primo comando da utilizzare nel caso che il file che trattiamo sia un nuovo file.
La sintassi del comando è:

lineaI

Se non indichiamo alcun numero di linea o scriviamo ".I" l'editor assume che l'inserimento inizi immediatamente prima della linea in corso di trattamento. Se il numero che indichiamo è superiore all'ultimo numero presente nel nostro documento, o indichiamo il simbolo "#" al posto del numero di linea, l'inserimento della nuova riga viene effettuato dopo l'ultima riga del nostro testo. Attivata la funzione di inserimento l'editor visualizza il numero della riga su cui operiamo, incrementandone il numero tutte le volte che premiamo il tasto ENTER. Per terminare la sessione di inserimento dobbiamo premere, come primo carattere di una linea, i tasti Ctrl e C (o Ctrl e Break o, ancora, Ctrl e Z). L'editor ripresenta l'asterisco a sinistra del nostro schermo restando in attesa di un nuovo eventuale comando. La linea immediatamente successiva all'ultima inserita diventa la linea corrente.
Un esempio chiarirà meglio il funzionamento. Riprendiamo il nostro testo:

```
      1:*   Linea di testo numero uno
      2:    Linea di testo numero due
      3:    Linea di testo numero tre
      4:    Linea di testo numero quattro
*
```

Vogliamo inserire alcune righe dopo la riga numero 2. Scriviamo quindi "3i". Apparirà:

```
      3:*
```

Digitiamo, ad esempio, A e ENTER, B e ENTER:

```
      3:A
      4:B
      5:*
```

Sulla riga numero 5 digitiamo Ctrl e C. Visualizziamo ora il testo modificato:

```
      1:    Linea di testo numero uno
      2:    Linea di testo numero due
```

```
    3:    A
    4:    B
    5:*   Linea di testo numero tre
    6:    Linea di testo numero quattro
*_
```

La linea puntata dall'editor è la numero 5 corrispondente alla nostra linea numero 3 prima degli inserimenti effettuati.

COMANDO "L" ("LIST", ELENCA)

Questo comando permette di visualizzare una o più linee. Le linee visualizzate restano invariate.
La sintassi del comando è:

```
linea, lineaL
```

Viene visualizzato l'intervallo di testo che va da linea a linea come da noi specificato. Se omettiamo il primo parametro indicando solo:

```
,lineaL
```

otteniamo la visualizzazione delle 11 linee precedenti la linea in corso di trattamento e delle linee successive fino al raggiungimento del secondo parametro da noi indicato. È obbligatoria l'indicazione della virgola prima del numero per indicare che il primo parametro è stato omesso.
Omettendo entrambi i parametri ci vengono visualizzate un numero massimo di 23 linee, undici prima e undici dopo la linea in corso. Se non esistono nel nostro testo undici righe prima di quella corrente, l'editor ci mostra le righe, successive del testo fino al numero massimo di 23 linee.
Possiamo servirci anche della numerazione relativa. Per esempio, il comando:

```
-2,+2L
```

ci mostra le due linee precedenti, la linea in corso di trattamento e le due linee successive.

COMANDO "M" ("MOVE", SPOSTAMENTO)

Questo comando permette di spostare una o più linee da una parte all'altra del nostro testo. Possiamo assimilare questo comando al comando di copia con la differenza che le linee copiate vengono eliminate dalla posizione originale e, inoltre, possiamo indicare un solo spostamento per volta.
La sintassi di questo comando è la seguente:

```
linea inizio,linea fine,linea destinazioneM
```

Il parametro linea inizio ci indica qual è la prima linea del nostro testo che intendiamo muovere. Il parametro linea fine ci indica l'ultima linea che intendiamo muovere. Se uno di questi due parametri viene omesso, l'editor attribuisce come linea inizio o linea fine il numero corrispondente alla linea in corso di trattamento. È invece obbligatorio il parametro linea destinazione per determinare a quale altezza del nostro testo vanno inserite le linee da muovere. L'editor inserisce quindi le linee copiate a partire dalla linea da noi indicata come linea di destinazione della copia. La prima delle linee inserite a destinazione diventa la linea corrente. Le linee vengono rinumerate tenendo conto della direzione di spostamento.

COMANDO "P" ("PAGE", PAGINA)

Questo comando ci permette di visualizzare il nostro testo una videata alla volta. La sintassi del comando è:

```
linea, lineaP
```

Se omettiamo il primo parametro, l'editor visualizza a partire dalla riga successiva a quella in corso di trattamento. Se omettiamo il secondo parametro, ci vengono visualizzate 23 linee. La linea che diviene corrente è l'ultima linea visualizzata dal comando P.
Sia con il comando L che con il comando P non viene alterato il contenuto delle linee visualizzate, l'unica differenza fra questi due comandi è che il comando P varia la linea corrente.

COMANDO "Q" ("QUIT", ABBANDONO)

Con questo comando usciamo dal programma EDLIN lasciando invariato (o inesistente) il nostro file. Per ulteriore conferma, l'editor pone la seguente domanda:

```
Do you want to end the edit (Y/N)?
```

Digitando "N" riprendiamo il trattamento del nostro file esattamente da dove abbiamo digitato Q senza che nulla sia cambiato, digitando "Y" tutto ciò che era stato eventualmente modificato viene perso e riappare il prompt del sistema operativo.

COMANDO "R" ("REPLACE", SOSTITUZIONE)

Questo comando permette di sostituire una sequenza di caratteri a un'altra, ripetendo la sostituzione per l'intervallo di linee indicate.
La sintassi del comando è:

```
linea inizio,linea fine?Rsequenza origF6sequenza sost
```

Se non indichiamo la sequenza sost di sostituzione, l'editor cancellerà tutti i caratteri indicati nella prima sequenza (sequenza orig) per tutte le linee da noi scelte.
Se omettiamo l'indicazione di entrambe le sequenze, l'editor utilizza quelle usate nell'ultimo comando R o S eseguito.
La dicitura F6 sta per la pressione del tasto F6, cui può essere sostituita la coppia di tasti Ctrl e Z e serve per notificare all'editor la fine della sequenza di caratteri.
L'editor visualizza le righe sullo schermo man mano che le modifica. Se nel comando non indichiamo il "?", la sostituzione avviene automaticamente per tutte le linee da noi scelte. L'indicazione del punto interrogativo provoca invece una richiesta di conferma della sostituzione, attraverso il prompt (OK?), per ogni riga trattata. Rispondendo "Y", l'editor effettuerà la sostituzione in quella riga, rispondiamo "N", o qualsiasi carattere diverso da "Y", non effettuerà la sostituzione in quella linea, proseguendo comunque alla successiva.
Se omettiamo il primo numero di linea, la sostituzione comincia dalla linea successiva a quella cui l'editor punta. Se omettiamo il secondo numero di linea, la sostituzione termina alla fine del nostro documento. Se omettiamo entrambi i parametri di delimitazione dell'intervallo di linee, la sostituzione comincia con la linea successiva a quella puntata dall'editor e termina con il raggiungimento della fine documento.
La ricerca della sequenza di sostituzione tiene conto anche dei caratteri maiuscoli e minuscoli, all'interno sia delle linee di testo che vengono trattate, che della sequenza da noi specificata.

COMANDO "S" ("SEARCH", RICERCA)

Questo comando permette la ricerca di una sequenza di caratteri nell'intervallo di linee da noi indicato.
La sintassi del comando è:

```
linea inizio, linea fine?Ssequenza
```

I parametri linea inizio e linea fine indicano all'editor su quale parte di testo deve essere svolta la ricerca.
Se omettiamo il parametro linea inizio, l'editor effettua la ricerca dalla prima linea successiva a quella identificata corrente.
Se, invece, omettiamo il parametro linea finale, l'editor prosegue la ricerca fino all'ultima linea del nostro documento.
Possiamo indicare come sequenza da ricercare un numero qualsiasi di caratteri, tenuto conto che la stringa viene ricercata esattamente come la indichiamo, rispettando, quindi, le eventuali maiuscole o minuscole presenti.
Se non specifichiamo nessuna sequenza, l'editor assume quella indicata nell'ultimo comando R o S eseguito.
Se la sequenza di caratteri da noi indicata non appare in nessuna delle linee specificate, l'editor visualizza il messaggio

```
Not found
```

Se non indichiamo il parametro "?" l'editor termina la ricerca alla prima linea che contiene la sequenza indicata. Per continuare la ricerca su più linee dobbiamo includere il punto interrogativo nella sintassi di immissione del comando. Come già visto per il comando R, il punto interrogativo obbliga l'editor a presentare la richiesta (OK?) ogni volta che la sequenza viene individuata nella linea in esame. Rispondendo "Y" la ricerca termina e la riga corrente diventa la linea di trattamento. Rispondendo "N", o qualsiasi altro carattere diverso da "Y", la ricerca prosegue secondo i parametri indicati.

COMANDO "T" ("TRANSFER", TRESFERIMENTO)

Questo comando permette di trasferire il contenuto di un altro file di testo nel file di testo che stiamo attualmente trattando.
La sintassi del comando è:

```
linea,T pathname
```

Linea, se specificata, indica il numero di linea da cui inizia l'inserimento del documento specificato da pathname. Le linee vengono rinumerate automaticamente dall'editor. Non indicando alcun numero di linea il file letto viene inserito a partire dalla linea corrente.
Il file che vogliamo inserire nel nostro documento deve esistere, altrimenti riceveremo un messaggio di errore.

COMANDO "W" ("WRITE", SCRITTURA)

Questo è il comando che ci permette di scrivere una o più di righe del documento che stiamo trattando in un altro file.
La sintassi del comando 6:

```
lineaW
```

Se linea non è specificata, verrà aggiornato su disco un numero di linee sufficiente a far sì che la memoria che rimane occupata non superi il 25% del totale disponibile. Le linee vengono memorizzate cominciando dalla linea contrassegnata dal numero 1.
Questo comando serve per liberare parte della memoria occupata dal file in corso di trattamento per potervi caricare la parte ancora residente su disco, non caricata all'avviamento di EDLIN perché troppo grande.
Prima quindi di usare il comando "A", dovremo avvalerci di questo comando per liberare parte della memoria e mettere il nostro calcolatore nella condizione di poter reperire il resto del documento.

UTILIZZO DEI TASTI FUNZIONE NEL PROGRAMMA EDLIN

Abbiamo accennato a qualche tasto funzione nell'esposizione dei comandi del programma EDLIN: in effetti, per questo comando, i tasti funzione si comportano esattamente come per i comandi digitati dal prompt del sistema operativo, per cui si rimanda al capitolo 4 e all'appendice A.

CAPITOLO SETTE

OS/2 lavora per noi: i file "batch" e il reindirizzamento dell'I/O

ELABORAZIONE "BATCH" ED ELABORAZIONE "INTERATTIVA"

In OS/2 è possibile utilizzare le risorse del calcolatore in due modi: il primo, detto "interattivo" per l'immediatezza con cui il risultato segue la richiesta, è quello realizzato dai comandi visti nel capitolo 5; il secondo, detto "batch (raggruppato)", si realizza raccogliendo, in un file di testo, una serie di comandi perché vengano eseguiti in sequenza in un secondo tempo.
Se, ad esempio volessimo, per ogni dischetto che formattiamo, eseguire la verifica dello spazio totale disponibile, dovremmo digitare prima il comando FORMAT e poi il comando CHKDSK. Lo stesso risultato può essere ottenuto inserendo i nomi dei due comandi in un file di testo realizzato con EDLIN e richiamando poi dal prompt del sistema operativo il nome di questo file.
Affinché OS/2 riconosca i file di comandi batch, occorre che questi abbiano una particolare estensione, differenziata a seconda dell'ambiente operativo:

CMD	è l'estensione per i file di comando batch eseguibili in modo protetto;
BAT	è l'estensione per i file di comando batch eseguibili in modo reale.

Per realizzare quanto sopra esposto, supponiamo, in modo reale, di creare il comando batch FORMATTA. Al prompt del sistema operativo dovremmo digitare:

```
EDLIN FORMATTA.BAT
```

seguito dal comando I per insert, poi da:

```
1:    FORMAT A:
2:    CHKDSK A:
3:    ^C
```

seguito infine da E per exit. A questo punto, digitando FORMATTA, il sistema operativo eseguirà in sequenza i comandi che compongono il nostro file di testo, trattando ogni riga del file come fosse un comando da noi digitato al prompt di OS/2.
Per creare il comando analogo per il modo protetto dobbiamo, sempre in modo reale perché EDLIN funziona solo in questo modo, creare il file FORMATTA.CMD ed eseguirlo poi in modo protetto. Un modo più rapido e altrettanto efficace è di copiare il file suddetto con un nome diverso:

```
COPY FORMATTA.BAT FORMATTA.CMD
```

È bene precisare che non è il modo in cui sono stati creati o il loro contenuto che rende eseguibili i file così fatti: è semplicemente una caratteristica di OS/2 il tentare di eseguire linea dopo linea il contenuto di un file con estensione .BAT in modo reale e .CMD in modo protetto. In altri termini, se creiamo con l'editor un file con una delle estensioni suddette contenente un testo generico (cioè non comandi) e, nell'ambiente opportuno, ne digitiamo il nome al prompt del sistema operativo, questo tenterà di eseguire le parole componenti il nostro testo come fossero comandi, dando naturalmente per ogni riga, anziché il risultato di un comando, un messaggio di errore.

UTILIZZO DEI PARAMETRI DELLA LINEA DI COMANDO

OS/2 fornisce un metodo per utilizzare in un file batch gli eventuali parametri forniti sulla linea di comando: nel testo del file è infatti possibile identificare questi argomenti con il loro numero di posizione preceduto dal segno "%". In altri termini, se il testo del nostro file FORMATTA viene variato così:

```
FORMAT %1:
CHKDSK %1:
```

questa versione, a differenza della precedente che operava solo sul drive A, può operare indifferentemente su qualsiasi drive del sistema, purché se ne specifichi il nome al momento in cui il comando viene richiamato. Per formattare e verificare il disco A occorrerà quindi digitare:

```
FORMATTA A
```

mentre per formattare e verificare il disco B:

```
FORMATTA B
```

Ciò che in pratica avviene è la sostituzione del parametro "%1" con il primo argomento che segue il nome del comando, cioè la lettera A. La parametrizzazione degli argomenti è posizionale: al nome del comando corrisponde %0, al primo argomento %1 e così via fino a %9, assumendo un argomento separato dal successivo da almeno uno spazio.

COME EVITARE LA PRESENTAZIONE DEI COMANDI IN ESECUZIONE

Normalmente, durante l'esecuzione di un comando batch, i comandi compaiono sul monitor a fianco del prompt prima di essere eseguiti: questo può essere evitato sia tramite l'uso del comando ECHO, che vedremo dopo, sia premettendo a ogni comando il carattere "@" nel file di testo. Se, per quanto riguarda l'esempio precedente, non volessimo veder comparire sul video i comandi FORMAT A: e CHKDSK A:, dovremmo variare il nostro FORMATTA in questo modo:

```
@FORMAT %1:
@CHKDSK %1:
```

Nel solo modo protetto, esiste anche un'opzione da digitare sulla linea di comando che ottiene lo stesso effetto: /Q (da quiet, silenzioso). È da notare che /Q può essere posto in qualsiasi posizione rispetto ad altri eventuali argomenti, in quanto viene interpretato dal sistema operativo prima di avviare il comando e pertanto non occupa nessun parametro posizionale.

COMANDI PER I FILE BATCH

Oltre a tutti i comandi già visti precedentemente, nei soli file batch è possibile utilizzare un altro piccolo gruppo di comandi che esponiamo di seguito con il medesimo formalismo usato per il capitolo 5:

CALL

Ambiente: *Entrambi*

Sintassi: CALL pathname [argomento][...]

Questo comando permette di far proseguire l'elaborazione con un altro file batch, per riprendere poi dall'istruzione seguente CALL al termine del file chiamato. Pathname individua il nome del file batch mandato in esecuzione e argomenti identifica appunto gli argomenti eventuali passati a detto file.

ECHO

Ambiente: *Entrambi*

Sintassi: ECHO [ON|OFF|messaggio]

Questo comando esegue due diverse funzioni: con i parametri ON o OFF stabilisce il modo di presentazione dei comandi successivi. ECHO OFF evita, a partire dal comando successivo, la comparsa del nome del comando prima della sua esecuzione (equivale all'opzione /Q vista in precedenza); ECHO ON provoca il comportamento opposto. ECHO senza alcun argomento mostra se lo stato della visualizzazione dei comandi è ON o OFF. Infine, seguito da un generico messaggio, lo mostra sul video, indipendentemente dagli altri fattori.

Comandi affini: REM, PAUSE

ENDLOCAL

Ambiente: OS/2

Sintassi: ENDLOCAL

Questo comando ripristina i valori che le variabili di ambiente avevano prima del comando SETLOCAL (vedi oltre). Ad ogni screen group è associato un "ambiente" costituito da un certo numero di valori, come ad esempio PATH, PROMPT, ecc. Se queste variabili vengono alterate all'interno di un file batch, la variazione effettuata rimane anche quando il comando è terminato: con il comando SETLOCAL, invece, è possibile creare una sorta di ambiente "locale" a cui fanno riferimento le assegnazioni successive fino appunto al comando ENDLOCAL.

Comandi affini: SETLOCAL

EXTPROC

Ambiente: OS/2

Sintassi: EXTPROC pathname [argomento][...]

L'esecuzione dei file batch è resa possibile dalla presenza del "command processor": in effetti è a carico di quest'ultimo la trasformazione della sequenza di comandi memorizzati nel file in una sequenza di comandi che appaiono a tutti gli effetti al sistema operativo come digitati dall'utilizzatore.
Il command processor attivo in modo protetto è CMD.EXE: tramite il comando EXTPROC è possibile utilizzare, per la sola durata dell'esecuzione del file batch, un altro command processor. A causa di quanto appena specificato, è indispensabile che questo comando si trovi nella prima linea del file batch, al fine di permettere a tutti i comandi successivi di subire un trattamento omogeneo da parte di un medesimo programma. Pathname è il nome del file contenente il command processor alternativo, argomenti sono chiaramente quelli che gli vengono eventualmente passati.

FOR

Ambiente: *Entrambi*

Sintassi: FOR %%variabile IN (elemento[...]) DO comando

Questo comando permette di ripetere il comando che segue il DO tante volte quanti sono gli elementi, assegnando ad ogni ciclo alla variabile il valore di un elemento. La variabile è costituita da un solo carattere, che può essere indifferentemente una lettera o un numero. I parametri posizionali, poiché sono preceduti da un solo "%", a differenza della variabile di un ciclo FOR, non provocano alcuna ambiguità. Vediamo un esempio:

```
FOR %%A IN (LETTERA1.TXT LETTERA2.TXT LETTERA3.TXT) DO DIR %%A
```

Il risultato di questo comando è il medesimo che si otterrebbe digitando:

```
DIR LETTERA1.TXT
DIR LETTERA2.TXT
DIR LETTERA3.TXT
```

È da notare che è possibile, proprio perché il command processor che esegue i file batch è il medesimo che presenta il prompt di comando, eseguire un comando come il FOR al prompt del sistema operativo; tuttavia in questo caso la variabile, sia dopo il FOR che dopo il DO, va indicata preceduta da un solo "%". Se digitato al prompt, l'esempio precedente diventa:

```
FOR %A IN (LETTERA1.TXT LETTERA2.TXT LETTERA3.TXT) DO DIR %A
```

GOTO

Ambiente: *Entrambi*

Sintassi: GOTO indirizzo

Grazie a questo comando è possibile far proseguire l'elaborazione del file batch con una linea che non sia necessariamente successiva a quella contenente il comando. Per specificare indirizzo, occorre inserire in una linea, all'inizio, il carattere ":" seguito immediatamente dal nome che caratterizza quel determinato indirizzo, che non deve contenere il ".". Se l'indirizzo specificato dopo GOTO non esiste, l'elaborazione del file batch termina.

Comandi affini: IF

IF

Ambiente: *Entrambi*

Sintassi: IF condizione comando

Questo comando permette di condizionare l'esecuzione di comando al verificarsi o meno di condizione. La condizione può essere specificata nel modo seguente:

ERRORLEVEL numero	La condizione è verificata se il programma precedente ha riportato un codice di terminazione maggiore o uguale a ERRORLEVEL. Il codice di terminazione è un numero di una cifra che ogni programma rende disponibile al successivo per comunicare la riuscita (ERRORLEVEL=0) o il fallimento del proprio scopo (ERRORLEVEL>0).
nome1 == nome2	La condizione è verificata se il contenuto delle due variabili di ambiente (o locali) è uguale.
EXIST nome file	La condizione è verificata se nella directory corrente esiste il file con il nome specificato. È possibile includere nel nome del file i caratteri "?" e "*".
NOT condizione	Premessa ad una qualsiasi delle condizioni precedenti, questa condizione è verificata in tutti i casi in cui le precedenti non lo erano.

Poiché il comando specificato viene eseguito solo quando la condizione è verificata, IF è utilizzato spesso per eseguire diverse sezioni di file batch associato al comando GOTO. Facciamo un esempio in cui l'argomento digitato determina la sezione di file da eseguire: supponiamo di creare un comando che cancella il file se esiste o manda un messaggio se non esiste. Questo file batch potrebbe essere così composto:

```
IF EXIST %1 GOTO CANC
ECHO Il file %1 non esiste
GOTO FINE
:CANC
DEL %1
ECHO Cancellato il file %1
:FINE
```

Supponendo di chiamare questo file CANCELLA, digitando CANCELLA PIPPO, se un file con questo nome esiste riceveremo il messaggio "Cancellato il file PIPPO", altrimenti riceveremo il messaggio "Il file PIPPO non esiste".

Comandi affini: GOTO

PAUSE

Ambiente: *Entrambi*

Sintassi: PAUSE [messaggio]

Questo comando permette di sospendere l'elaborazione di un file batch: sul monitor compare il messaggio

"Press any key when ready"

e, alla pressione di un tasto qualsiasi, l'elaborazione riprende dal comando successivo a PAUSE. Se messaggio è stato specificato, viene presentato prima il messaggio e subito dopo l'invito a premere un tasto per proseguire.

Comandi affini: ECHO, REM

REM

Ambiente: *Entrambi*

Sintassi: REM [messaggio]

Questo comando permette di inserire spazi o commenti nel testo del file batch. Se ECHO è OFF (o è stato impostata l'opzione /Q) questi commenti vengono semplicemente trascurati, viceversa, se ECHO è ON, ogni REM viene visualizzato con l'eventuale messaggio.

Comandi affini: ECHO, PAUSE

SETLOCAL

Ambiente: OS/2

Sintassi: SETLOCAL

Questo comando, già esaminato a proposito di ENDLOCAL, stabilisce da quale punto del file batch le assegnazioni di eventuali variabili rimangono locali al file e non alterano i valori di eventuali altre variabili di ambiente che hanno il medesimo nome.

Comandi affini: ENDLOCAL

SHIFT

Ambiente: *Entrambi*

Sintassi: SHIFT

Questo comando permette di utilizzare i parametri forniti sulla linea di comando del file batch quando questi sono più di 10. Poiché non è possibile riferirsi a parametri posizionali che occupino una posizione superiore alla nona, occorre inserire uno SHIFT all'interno del file batch per avere uno spostamento verso sinistra della numerazione dei parametri rispetto alla loro posizione originale. In pratica, ogni volta che viene effettuato uno SHIFT, il valore che precedentemente era individuato da %2 viene a trovarsi in %1 e il valore che precedentemente era in %0 viene perso. Naturalmente non è affatto obbligatorio utilizzare il comando SHIFT solo quando gli argomenti sono più di 10: questo comando può essere utilizzato all'interno di un ciclo per utilizzare sempre il medesimo numero di argomento, cambiandone di volta in volta il valore con il comando SHIFT.

ALCUNI FILE NOTEVOLI: STARTUP.CMD, OS2INIT.CMD E AUTOEXEC.BAT

Ci sono alcuni file "batch" che rivestono particolare importanza per il sistema operativo:

OS2INIT.CMD è il file batch che viene eseguito ogni volta che viene attivata una nuova sessione in modo protetto.

AUTOEXEC.BAT è il file batch che viene eseguito all'attivazione della sessione di emulazione DOS (in effetti riveste la medesima funzione del file AUTOEXEC.BAT del DOS, con l'unica differenza che non viene eseguito al momento dell'accensione del calcolatore, ma la prima volta che dal program selector viene avviata la sessione in modo reale).

STARTUP.CMD Se questo file è presente nella directory root del disco da cui viene avviato il calcolatore, il program selector attiva automaticamente una sessione in modo protetto che inizia con l'esecuzione del file batch suddetto; altrimenti, se STARTUP.CMD non esiste, o non si trova nella directory root del disco suddetto, sul monitor compare il program selector.

La procedura di installazione crea i file OS2INIT.CMD e AUTOEXEC.BAT, inserendo in entrambi la definizione del PATH di ricerca comandi (capitolo 5), così come risulta dall'installazione, e abilita la funzione di HELP (capitolo 5) che appare sulla prima linea del monitor. Il file STARTUP.CMD può o meno essere creato al momento dell'installazione, nel caso che lo sia, non fa altro che eseguire a sua volta OS2INIT.CMD in modo che la sessione attivata automaticamente non manchi dei parametri fondamentali.
Tipicamente, il contenuto di OS2INIT.CMD e AUTOEXEC.BAT al termine dell'installazione è il seguente:

```
@echo off
PATH=c:\;c:\os2;c:\os2\install
CALL HELP
```

Notiamo l'istruzione CALL HELP: il comando HELP infatti, descritto al capitolo 5, è in realtà costituito attualmente da due file batch, uno per ciascun modo operativo (HELP.CMD e HELP.BAT), che invocano a loro volta il comando HELPMSG. Non è escluso che future versioni di OS/2 sostituiscano questa procedura con un comando HELP vero e proprio, tuttavia essa rappresenta un ottimo esempio di come uno stesso comando possa essere utilizzato con modalità diverse e fini simili da due file batch in modo reale e protetto.Un'altra importante osservazione da fare riguarda il diverso modo in cui il file di "startup" è trattato dai due diversi "command processor" (vedi capitolo 5, CMD.EXE e COMMAND.COM): la semplice

esistenza di un file chiamato AUTOEXEC.BAT nella directory root del disco di partenza porta il command processor del DOS ad eseguirlo non appena viene attivato per la prima volta; viceversa, il command processor di OS/2 esegue il file OS2INIT.CMD se questo è specificato in CONFIG.SYS, comando PROTSHELL (capitolo 9), e questo presenta l'evidente vantaggio di poter dare al file di startup di OS/2 anche un altro nome.

IL TRATTAMENTO DELL'INPUT/OUTPUT: STDIN, STDOUT E STDERR

Le parole INPUT e OUTPUT significano, rispettivamente, INGRESSO e USCITA, tuttavia è diventato talmente comune parlare dell'I/O che anche in questo caso useremo la terminologia anglosassone.
Qualsiasi programma sia eseguito su un calcolatore, compreso il sistema operativo stesso, interagisce con tutti i dispositivi, sia quelli interni al calcolatore (come memoria RAM, orologio interno, dischi, "porte" di comunicazione seriali e parallele) che quelli esterni (tastiera, video, mouse, stampanti ecc.) tramite dei "canali" di I/O. In altri termini, per ogni operazione che comporta un trasferimento di dati da un punto all'altro del calcolatore, occorre sia disponibile un canale, solitamente atto alla comunicazione nei due sensi (appunto INGRESSO e USCITA), perché questa comunicazione possa avvenire. È da notare che non sempre questo canale è rappresentabile con un qualche oggetto, sia esso reale o figurato: il trasferimento dei dati dai dischi alla memoria RAM del calcolatore e viceversa, ad esempio, avviene attraverso uno di questi canali, che malgrado abbia un nome, DMA, non è collegabile a nessun particolare osservabile del calcolatore, in quanto è completamente interno alla costruzione dello stesso.
Tre di questi canali, che si chiamano STANDARD INPUT, STANDARD OUTPUT e STANDARD ERROR ci interessano particolarmente perché, grazie alla possibilità di "reindirizzarli", cioè di cambiare la loro destinazione, permettono di estendere il campo di azione dei file batch.

- Lo STANDARD INPUT (abbreviato STDIN) è il canale, in cui tra l'altro i dati viaggiano in un solo senso come suggerisce anche il nome, tramite cui un programma riceve i dati richiesti durante l'esecuzione, come ad esempio la risposta ad una richiesta di conferma, la data e l'orario corrente e, in generale, tutto ciò che va fornito ad un programma mentre l'elaborazione è in corso: normalmente tale canale è connesso alla tastiera e al mouse del nostro sistema.
- Lo STANDARD OUTPUT (abbreviato STDOUT) è il canale, anche questo percorso in un solo senso, utilizzato dai programmi per inviare messaggi, presentare maschere video, richieste di conferma, e, in generale, tutto ciò che, durante il loro normale funzionamento, va esposto: normalmente tale canale è connesso al monitor del nostro sistema.
- Lo STANDARD ERROR (abbreviato STDERR) è il canale tramite cui, analo-

gamente al precedente, alcuni messaggi vengono inviati perché siano visualizzati: la differenza è che su questo canale i programmi inviano, solitamente, solo i messaggi di errore. Anche questo canale è normalmente connesso al monitor.

Questi tre canali, che sono associati ad ogni programma, compreso il sistema operativo stesso, sono caratterizzati da tre numeri che, come vedremo, possono essere utilizzati, nel solo modo protetto, per il loro trattamento nei programmi:

0	(STDIN)	STANDARD INPUT
1	(STDOUT)	STANDARD OUTPUT
2	(STDERR)	STANDARD ERROR

I concetti che abbiamo esposto divengono più chiari se pensiamo che, ad esempio, l'operazione di visualizzazione della directory di un disco, che possiamo eseguire con il comando DIR (capitolo 5), altro non è che la trasmissione, da parte del comando DIR, dei nomi letti sul disco nel proprio canale STDOUT.
Analogamente, la richiesta di conferma che il comando DEL (capitolo 5) ci pone se, ad esempio, digitiamo DEL *.*, altro non è che la trasmissione, da parte del comando DEL, della sequenza

```
Are you sure (Y/N)?
```

sul proprio STDOUT, seguita dalla ricezione del carattere corrispondente al tasto da noi premuto, dal proprio STDIN.

IL REINDIRIZZAMENTO DI STDOUT, STDIN E STDERR

In OS/2 esiste la possibilità, per ciascuno di questi canali, di cambiare temporaneamente l'assegnazione e di spostarla, dal dispositivo standard ad un file o ad un altro dispositivo. Questa possibilità, che va sotto il nome di reindirizzamento dell'I/O, viene effettuata tramite una serie di operatori:

<pathname	Anziché collegare STDIN (canale 0) dalla tastiera, viene letto il file o il dispositivo individuato da pathname.
>pathname	Anziché collegare STDOUT (canale 1) al monitor, i messaggi vengono scritti nel file o nel dispositivo individuato da pathname.
>>pathname	Come nel caso precedente, con la differenza che, se il file individuato da pathname esisteva già, i messaggi provenienti da STDOUT vengono aggiunti alla fine del file.
<&n	*n* individua il numero del canale che viene "duplicato" per essere utilizzato come STDIN. n può essere 0, 1 o 2.

>&n — n individua il numero del canale che viene "duplicato" per essere utilizzato come STDOUT. n può essere 0, 1 o 2.

n> — n individua il numero del canale che viene reindirizzato: in questo caso, n può essere 1 o 2.

L'utilizzo degli ultimi tre operatori è riservato al modo protetto. Per quanto riguarda la sintassi, occorre notare che non devono assolutamente esservi spazi fra i numeri dei canali, i segni di reindirizzamento e "&".
Alcuni esempi chiariranno meglio il funzionamento di questi operatori. Se, ad esempio, noi volessimo stampare il contenuto della nostra directory corrente, potremmo utilizzare il reindirizzamento di STDOUT sulla stampante parallela LPT1, digitando:

```
DIR > LPT1:
```

tramite questo comando, anziché comparire sul video, l'indice dei file contenuti nella nostra directory corrente finisce stampato.
Un altro esempio può essere realizzato così: supponiamo di avere due file di testo, GEN_GIU.TXT e LUG_DIC.TXT, da unificare in uno, accodando il secondo al primo e realizzando il file GEN_DIC.TXT. In realtà, sappiamo già eseguire questa operazione tramite il comando COPY (capitolo 5), ma vediamo la differenza. Digitando:

```
TYPE GEN_GIU.TXT > GEN_DIC.TXT
```

creiamo il file GEN_DIC.TXT che peraltro, è una copia conforme di GEN_GIU.TXT.
Se adesso digitiamo:

```
TYPE LUG_DIC.TXT >> GEN_DIC.TXT
```

il file LUG_DIC.TXT viene accodato alla fine di GEN_DIC.TXT a formare un unico file che è l'unione dei precedenti.
Per fare un esempio sul funzionamento del reindirizzamento di STDIN, utilizziamo l'opzione /P del comando DIR, eseguito su una directory con molte voci (come, ad esempio, OS2). Tramite EDLIN, possiamo costruire un file che contenga semplicemente delle linee vuote, ottenute premendo, in fase di inserimento, più volte il tasto ENTER. Supponiamo di aver chiamato questo file ENTER; digitando:

```
DIR/P < ENTER
```

vedremo la directory scorrere sul video senza attendere le nostre conferme ad ogni schermata: questo perché le conferme vengono prese dal file ENTER. In effetti, se blocchiamo lo scorrimento premendo il tasto ScrLck, vedremo sul monitor, fra i nomi dei file, la dicitura:

```
Press any key when ready
```

che abbiamo visto nel comando DIR (capitolo 5), senza che tuttavia il calcolatore si sia fermato e ci abbia permesso di premere alcun tasto.
Il reindirizzamento dell'I/O è molto utile quando si vogliono far eseguire in "background", ad esempio tramite il comando DETACH (capitolo 5), programmi che normalmente richiedono dati da tastiera e mostrano messaggi sul monitor. Supponiamo di avere un programma del genere, chiamato AGGIORNA, il cui compito sia quello di verificare se i file relativi ad una determinata applicazione sono stati aggiornati con i dati di file relativi ad un'applicazione collegata alla prima. Quando viene eseguito richiamandolo dal prompt del sistema operativo, AGGIORNA richiede una coppia di nomi di file su cui verificare l'aggiornamento, e, dopo averlo eseguito, segnala quanti sono stati gli aggiornamenti fatti. Poiché questo programma va ripetuto per numerose coppie e i nomi dei file sono sempre i medesimi, abbiamo deciso di eseguirlo in "background": creiamo con EDLIN tanti file quante sono le coppie di file da aggiornare, supporremo per brevità solo 2, e chiameremo quindi i due file, ad esempio, MAGAZZI e BOLLE. Dopodiché, sempre con EDLIN, creeremo il file AGG_FILE.CMD che conterrà le linee seguenti:

```
AGGIORNA < MAGAZZI
AGGIORNA < BOLLE
```

e infine, digiteremo al prompt del sistema operativo:

```
DETACH AGG_FILE > AGG_FILE.TXT 2>&1
```

In questo modo, il programma AGGIORNA, richiamato per due volte consecutive, riceverà i nomi di ciascuna coppia di file dal file da noi predisposto anziché dalla tastiera; e i dati relativi all'aggiornamento verranno globalmente raccolti nel file AGG_FILE.TXT. Notiamo che anche gli eventuali messaggi di errore vengono memorizzati nello stesso file: la dicitura 2>&1 indica che STDERR viene "duplicato" e mandato su STDOUT che, a sua volta, è stato reindirizzato su AGG_FILE.TXT. Notiamo che l'esempio è stato fatto supponendo di essere in modo protetto (estensione CMD del file batch): in effetti, in emulazione DOS, non è possibile reindirizzare STDERR (cioè non è ammessa la dicitura 2>&1) e, inoltre, non è possibile attivare alcun comando in background.
Il reindirizzamento dell'I/O è particolarmente interessante nella costruzione di filtri. Supponiamo di voler avere l'elenco dei file della nostra directory corrente con file i in ordine alfabetico: sappiamo già come indirizzare lo STDOUT del comando DIR su file; abbiamo visto che il comando SORT (capitolo 5) ordina i dati presenti sul proprio STDIN e li trasmette, ordinati, sul proprio STDOUT. Potremmo quindi digitare:

```
DIR > DIR_TEMP
```

```
SORT < DIR_TEMP
```

In effetti otterremmo quanto desiderato, tuttavia abbiamo dovuto creare un file intermedio, che andrà quindi cancellato, e, inoltre, abbiamo dovuto scrivere due comandi. Esiste, in effetti, un operatore prezioso per problemi come il precedente: "|", che si chiama *pipe* (abbreviazione anglosassone di *pipeline*, una tubatura composta da molti componenti) e il cui compito è quello di trasformare lo STDOUT del comando alla sua sinistra in STDIN per il comando all sua destra. Il nostro esempio precedente diviene:

```
DIR | SORT
```

Il risultato che vedremo sul monitor è esattamente equivalente a quello del caso precedente, però ottenuto senza creare alcun file e con un solo comando.
Non esiste limite al numero di "|" che si possono mettere in un comando: supponiamo di volere la lista alfabetica delle directory poste al livello della directory root del nostro disco. Abbiamo bisogno di un filtro che selezioni da STDOUT del comando DIR solo le linee con la dicitura "<DIR>" e poi passarle al comando SORT. Il comando che fa per noi è FIND, e il comando avrà questo aspetto:

```
DIR \ | FIND "<DIR>" | SORT
```

Se infine, volessimo stampare questo elenco sulla stampante connessa come LPT1, dovremmo digitare:

```
DIR \ | FIND "<DIR>" | SORT > LPT1:
```

IL RAGGRUPPAMENTO DEI COMANDI E L'ESECUZIONE CONDIZIONATA

Per il solo modo protetto, esistono alcuni operatori che permettono di rendere ancora più duttile il nostro sistema operativo. Questi operatori permettono di scrivere più comandi sulla stessa linea e di condizionare l'esecuzione di uno di essi al risultato del precedente. Se ad esempio abbiamo il dubbio che un determinato file di testo esista, prima di poter eseguire il comando TYPE dobbiamo verificarne l'esistenza tramite il comando DIR. Se supponiamo che il file sia RAPPORTO.TXT, il simbolo && (equivalente all'AND logico) ci permette di scrivere:

```
DIR RAPPORTO.TXT && TYPE RAPPORTO.TXT
```

Se RAPPORTO.TXT non esiste, il comando a destra di && non viene eseguito; viceversa, il contenuto di RAPPORTO.TXT viene mostrato sul video.
Se, invece, non ricordiamo più se ARCHIVIO sia un file o una directory ma comunque vogliamo cancellarlo, possiamo utilizzare il simbolo || (equivalente

all'OR logico) e scrivere:

```
DEL ARCHIVIO || RMDIR ARCHIVIO
```

Se ARCHIVIO è un file, allora il comando DEL riesce, pertanto il comando RMDIR non viene eseguito; viceversa, se ARCHIVIO è una directory, viene eseguito il comando RMDIR. L'operatore & permette di scrivere più comandi sulla stessa linea che verranno eseguiti uno di seguito all'altro, come se fossero stati digitati uno per volta. Infine, l'operatore ^ permette di includere in un testo uno dei segni fin qui visti (compresi quelli di reindirizzamento dell'I/O e il |) cancellandone, localmente, il significato speciale.
Esempi:

```
DIR A: & DIR C:
```

equivale a visualizzare la directory corrente del disco A e, di seguito, quella del disco C.

```
ECHO BUONGIORNO ^> ENFASI
```

non produce un file di nome ENFASI che contiene la parola BUONGIORNO, cosa che accadrebbe, per quanto detto più sopra, se non vi fosse il segno ^, ma visualizza il seguente messaggio:

```
BUONGIORNO > ENFASI
```

Se una linea di comando contiene più simboli riguardanti il reindirizzamento dell'I/O, |, && e ||, & e ^ OS/2 li valuta secondo un ordine di priorità che è il seguente, e che comprende anche le parentesi che possono venir utilizzate per variarlo:

```
^,(),simboli di reindirizzamento dell'I/O,|,&&,||,&
```

Questo significa che se noi vogliamo eseguire una coppia di comandi legati da & e, successivamente, reindirizzare STDOUT tramite | al SORT, dobbiamo racchiudere i primi due comandi fra parentesi. In effetti

```
DIR A: & DIR C: | SORT
```

ordina soltanto la directory corrente del disco C, mentre

```
(DIR A: & DIR C:) | SORT
```

ordina le directory di entrambi i dischi. L'unico carattere che prevale sulle parentesi è, ovviamente, ^ così da permettere di togliere il significato speciale

anche ad esse. Occorre notare che, se avessimo bisogno di includere il carattere ^ nei nostri testi, dovremmo digitare ^^.

CAPITOLO OTTO

Come installare OPERATING SYSTEM/2

DISTRIBUZIONE E INTALLAZIONE DI OS/2

Operating System/2 viene distribuito da entrambe le sorgenti, Microsoft e IBM, con un proprio programma di installazione. Il prodotto consiste di:

- un manuale utente (User guide);
- un set di dischi da 5 pollici per l'installazione di OS/2 su Personal Computer IBM AT o computer totalmente compatibili;
- un set di dischi da 3 pollici e 1/2 per l'installazione di OS/2 su Personal System/2.

L'intero sistema operativo, compresi tutti i programmi di utilità, risiede su tre dischi: il programma di installazione può trovarsi su uno di questi o in un quarto disco.
L'installazione di OS/2 è quanto mai semplificata dal programma che guida l'utilizzatore all'esecuzione dei passi necessari al trasferimento del sistema operativo sul disco rigido. Per iniziare, è sufficiente inserire il disco etichettato "Installation disk" nel drive A e riavviare il sistema.
Nel caso il calcolatore non sia dotato di una sufficiente quantità di memoria RAM (il minimo richiesto è 1,5 Mbyte) per avviare OS/2, sul video appare una breve segnalazione della versione di OS/2, subito seguita da un segnale acustico e dalla comparsa del messaggio:

The system does not have enough storage to
start *sigla costruttore* Operating System/2

The system is stopped

Correct the preceding error and restart the system

Il messaggio, che indica appunto che la memoria è insufficiente, segnala che il sistema è stato arrestato e che, "correggendo l'errore precedentemente segnalato si può riavviare il computer": naturalmente per correggere detto errore è necessario acquistare una espansione di memoria RAM. Notiamo che sigla costruttore può essere Microsoft o IBM. Se la memoria è sufficiente, appena avviato, il sistema presenta un "logo": appare cioè il marchio del costruttore del sistema operativo stesso. Questi, come abbiamo già detto, può essere indifferentemente Microsoft o IBM, in quanto il prodotto, realizzato dalla prima casa, viene poi adattato dalla seconda alle particolarità dei propri elaboratori (ciascuna casa ha, di comune accordo con l'altra, registrato il marchio OS/2 premettendogli la propria sigla: infatti dovremmo parlare, in realtà, di MS OS/2 e IBM OS/2, tuttavia le differenze, sebbene consistenti a livello dell'interfaccia del sistema verso l'hardware, sono praticamente nulle a livello di interfaccia verso l'utilizzatore).
Dopo la presentazione del logo, il programma di installazione verifica la configurazione del nostro calcolatore (numero di floppy disk drive, di hard disk drive, memoria RAM, numero di porte seriali e parallele, tipo di monitor, ecc.) e di conseguenza propone una configurazione di default dei parametri del file CONFIG.SYS (capitolo 9): accettando questa configurazione l'installazione prosegue automaticamente segnalando la copia dei file dal disco di installazione al disco rigido. Nel caso intendessimo variare qualche parametro durante la fase di installazione, dovremo intervenire su quelli che il programma di installazione ci permette di variare, oppure terminare l'installazione e variare il file CONFIG.SYS in un secondo tempo.
Se il programma di installazione rileva una quantità di spazio libero su disco rigido insufficiente alla copia dei file componenti OS/2 (il minimo richiesto è di 3 Mbyte), viene presentato un messaggio di errore che segnala questo fatto e il programma termina.
Quando il disco di installazione ha terminato la propria funzione, arresta il calcolatore segnalando che deve essere tolto dal drive e che l'elaboratore va riavviato premendo contemporaneamente i tasti Ctrl, Alt e Del (Canc nella tastiera italiana). Effettuando questa operazione, il calcolatore riparte con OS/2 e continua la fase di installazione dal disco rigido: i tre dischi contenenti Operating System/2 vengono richiesti dal sistema uno alla volta e i file in essi contenuti vengono copiati sul disco rigido. La progressione delle copie viene mostrata sul video, e, al termine di ogni disco, il programma di installazione chiede l'immissione del successivo e la pressione di un tasto per continuare. Al termine, il programma di installazione arresta nuovamente il calcolatore e ne richiede l'avviamento, dopo avere rammentato di togliere l'ultimo disco dal drive, tramite la medesima combinazione di tasti: dopo questa operazione l'installazione è terminata e il nostro Personal Computer è in grado di utilizzare OS/2.

INSTALLAZIONE DI OS/2 E CONVIVENZA CON IL DOS

Per coloro che posseggono già un elaboratore su cui sia installato il DOS, la procedura di installazione richiede alcune osservazioni preliminari: l'installazione di OS/2 avviene copiando una parte di file nella directory root del disco rigido, e creando due directory OS2 e SPOOL, entrambe a livello della directory suddetta. Tutti i programmi di utilità, i cui nomi potrebbero entrare in conflitto con alcuni programmi del DOS, vengono copiati, durante l'installazione, nella directory \OS2, pertanto l'unico problema di conflitto può nascere con file CONFIG.SYS che contiene i comandi di configurazione e che ha il medesimo nome sia in DOS che in OS2: la soluzione più rapida e indolore a questo problema è farsi, prima di installare OS/2, un disco DOS completo di sistema operativo (creato cioè con il comando FORMAT drive: /S del DOS), su cui copiare il CONFIG.SYS del DOS e l'AUTOEXEC.BAT e, in quest'ultimo file, inserire il comando:

```
SHELL=command.com c:\dos /p
C:
```

che farà si che, pur avviando il calcolatore da disco floppy, la nostra directory corrente sia C:\ e il sistema operativo sappia dove reperire la parte transiente del command processor COMMAND.COM (vedi capitoli 5 e 9) quando, al termine della procedura di partenza in DOS, toglieremo il disco dal drive A. Notiamo che abbiamo previsto che sul disco rigido esista una directory DOS (così come esisterà una directory OS2 al termine dell'installazione di quest'ultimo): l'esistenza di questa directory non è obbligatoria, tuttavia raccogliere i programmi di utilità del DOS e i vari device driver sotto un'unica directory rende molto più semplice ed ordinata la lettura del disco. Attenzione in quest'ultimo caso alle linee DEVICE nel CONFIG.SYS del DOS: andranno probabilmente corrette per indicare il path in cui cercare il device driver da installare.
Il file CONFIG.SYS viene comunque salvato dal programma di installazione con il nome CONFIG.BAK, questo per evitare che una mancata considerazione preliminare sulle problematiche poste dalla coesistenza di OS/2 e DOS sul medesimo disco produca effetti indesiderati. è possibile che in futuro venga offerto una sorta di "avviamento intelligente": la possibilità cioè che, appena acceso, il calcolatore richieda quale sistema operativo si intende utilizzare.

INTRODUZIONE A OS/2

Il programma di installazione prepara una directory, posta all'interno della directory OS2, chiamata INTRO: questa directory contiene un semplice programma di introduzione all'uso di OS/2, del program selector e dei principali comandi interni.
Utilizzare questo programma è semplicissimo, in quanto il programma di instal-

lazione ha preparato una sessione per esso nel program selector che appare la prima volta che OS/2 viene attivato al termine dell'installazione: basta muovere l'indicatore con i tasti direzionali o il mouse su questa voce, che figura nel riquadro intitolato "Start a program", e premere ENTER: verrà avviata una sessione con il programma dimostrativo suddetto e, al termine riapparirà il program selector.

INSTALLAZIONE DI OS/2 SU DISCO VERGINE

Se il disco rigido su cui intendiamo installare OS/2 non è mai stato utilizzato, il programma di installazione, prima di procedere alla copia dei file di OS/2 su detto disco, provvede a formattarlo. Questa operazione si avvale dei comandi FDISK e FORMAT, entrambi visti al capitolo 5; esponiamo di seguito la procedura per chi volesse realizzarla autonomamente o non disponesse del programma di installazione. Per eseguire questi comandi, è necessario avviare il sistema con il disco contenente il sistema operativo OS/2 nel drive A (questo disco dovrà naturalmente contenere anche i comandi FDISK e FORMAT). Il programma FDISK, quando viene richiamato, presenta una maschera video come quella di figura 8.1 (tenendo presente che sigla costruttore può essere Microsoft o IBM).

```
Personal Computer
Fixed Disk Setup Program Version 1.00

FDISK options


Choose one of the following:

    1.  Create a Microsoft Operating System/2 partition
        or a logical drive
    2.  Change the active partition
    3.  Delete a Microsoft Operating System/2 partition
        or a logical drive
    4.  Display the partition data


Enter choice: [1]



Press Enter to continue or
Esc to return to Microsoft Operating System/2
```

Figura 8.1

```
Create a Microsoft Operating System/2 partition

Choose one of the following:

    1.  Create primary Microsoft Operating System/2 partition
    2.  Create extended Microsoft Operating System/2 partition

Enter choice: [1]

Press Esc to return to the FDISK Options.
```

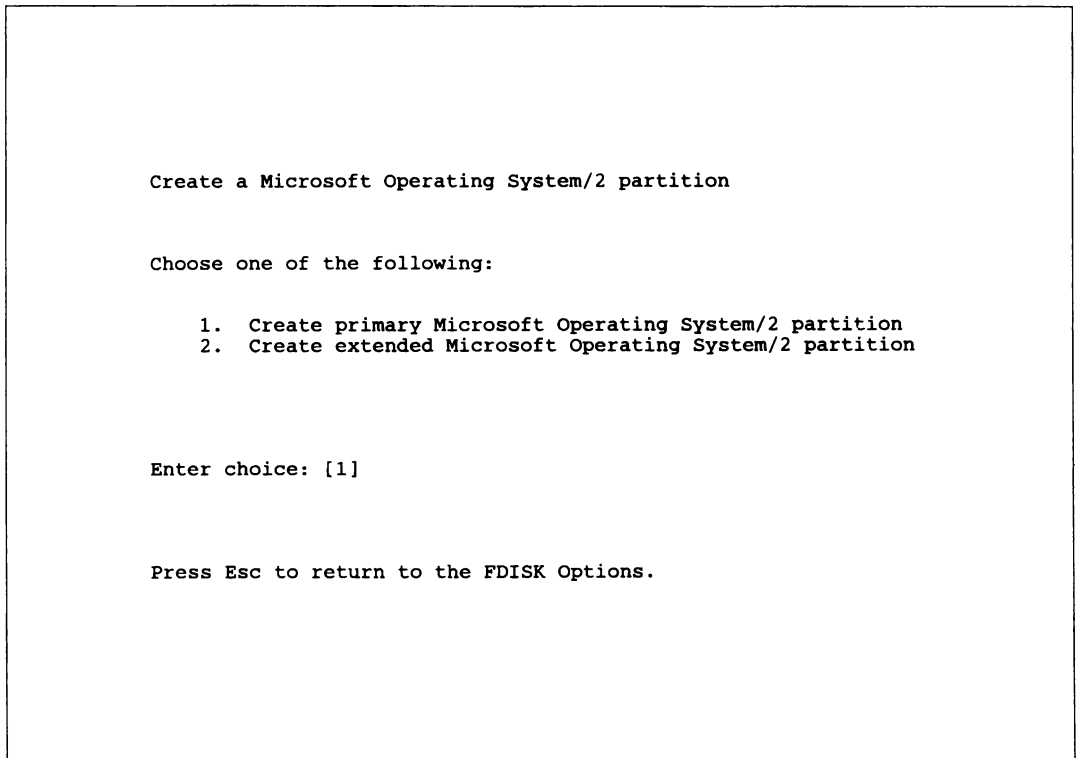



Figura 8.2

Noi sceglieremo l'opzione 1 (Creare una partizione OS/2), e, al menu della figura 8.2.
Sceglieremo ancora l'opzione 1 (Creare la partizione OS2 principale): a questo punto, il programma ci mostra il numero totale di cilindri del nostro disco rigido e il numero massimo di cilindri assegnabili alla partizione principale (se il disco è molto grande, questo numero sarà più piccolo del totale dei cilindri, altrimenti tutto il disco sarà proposto per la partizione). Una volta definito il numero del cilindro a cui far terminare la partizione principale, il comando FDISK arresta il sistema che va riavviato, ancora con il disco inserito nel drive A, premendo i tasti Ctrl, Alt e Del (o Canc) contemporaneamente. A questo punto, dopo l'avvio del sistema, occorrerà eseguire il comando:

```
FORMAT C: /S
```

che, dopo averci richiesto più volte conferma della nostra decisione in quanto questa operazione cancella completamente il contenuto del disco, eseguirà la formattazione di tutti i cilindri e, infine, la copia del sistema operativo e di tutti i file elencati nel file FORMATS.TBL, contenuto nel disco A, sul disco rigido appena formattato.

A questo punto, il disco floppy può essere tolto e il calcolatore può essere riavviato: OS/2 verrà caricato dal disco rigido e l'installazione potrà essere terminata manualmente, creando le directory \OS2 e \SPOOL e copiando il contenuto dei dischi restanti nella prima; dopodiché occorrerà soltanto personalizzare il file CONFIG.SYS.

CAPITOLO NOVE

Personalizzazione della configurazione di OS/2: CONFIG.SYS

La procedura di installazione, oltre a ricopiare OS/2 dai dischi su cui è distribuito sul disco rigido, inserisce nel file CONFIG.SYS alcuni parametri i cui valori vengono determinati durante la procedura stessa.
Il file CONFIG.SYS viene interpretato dal sistema operativo immediatamente dopo l'accensione del calcolatore e prima che venga eseguito qualsiasi comando e che appaia qualsiasi messaggio sullo schermo. Esso contiene tutta una serie di direttive che vanno dalla specifica della nazione al caricamento di programmi per l'utilizzo di dispositivi hardware particolari.CONFIG.SYS può essere gestito tramite l'editor EDLIN (capitolo 6) o con qualsiasi editor che però non inserisca alcun carattere di controllo nel testo. Le linee che costituiscono il file hanno tutte il formato:

```
COMANDO=parametri
```

dove parametri indica un generico valore che può essere una o più cifre, indicatori, nomi di file, ecc.
Anche per i comandi di CONFIG.SYS vale la distinzione fra DOS e OS/2: alcune delle direttive impartite in questo file hanno effetto solo quando ci si trova in emulazione dell'ambiente DOS, altre hanno effetto solo quando ci si trova in ambiente OS/2, altre ancora, infine, valgono in entrambi i casi.
Per tutti i comandi che possono essere dichiarati in CONFIG.SYS esiste un valore di default, assegnato dal sistema operativo nei tre casi seguenti:

- non esiste alcun CONFIG.SYS;
- un comando non compare in CONFIG.SYS;
- la sintassi di un comando specificato in CONFIG.SYS è errata.

Per ogni comando verrà specificato nome, ambiente, sintassi, default, funzione ed eventuali riferimenti a comandi correlati. Per ambiente s'intende OS/2, DOS o entrambi; l'esposizione della sintassi segue le medesime norme già specificate per quella dei comandi (capitolo 5).

BREAK

Ambiente: DOS

Sintassi BREAK={ON|OFF}

Default: BREAK=OFF

Questa comando istruisce OS/2, per il solo ambiente di emulazione DOS, su come trattare la richiesta di interruzione del programma in corso, effettuata dall'utilizzatore con la pressione contemporanea tasti CTRL e BREAK (o CTRL e C). Se BREAK=OFF, OS/2 prende in considerazione (e soddisfa) la richiesta solo se il programma da interrompere sta utilizzando il video o è in attesa di dati da tastiera e, in caso contrario, la soddisfarà non appena si presenta una di queste due condizioni. Se BREAK=ON, OS/2 soddisfa immediatamente la richiesta interrompendo il programma indipendentemente dall'uso che questo sta facendo delle risorse del sistema. Porre a ON il valore di BREAK può essere utile, ad esempio, per interrompere una lunga compilazione in quanto, poiché durante l'intero processo il programma compilatore non mostra messaggi a video e non necessita di dati da tastiera, risulterebbe impossibile da arrestare.

BUFFERS

Ambiente: *Entrambi*

Sintassi: BUFFERS=numero

Default: BUFFERS=3

Per buffer s'intende un'area di memoria RAM riservata alla memorizzazione temporanea di qualcosa che avrà poi un'altra destinazione. Poiché la memoria RAM è il più veloce dispositivo di memorizzazione, l'utilizzo di buffer per varie funzioni è molto diffuso (ad esempio, poiché una stampante è generalmente molto lenta nel trasformare un file di testo in un testo stampato, detto file viene solitamente memorizzato in un buffer più o meno grande presente solitamente all'interno della stampante stessa). In questo caso, il numero di *buffer* che andiamo a definire, ciascuno della dimensione di 512 byte, stabilisce quanta memoria RAM allocare per le funzioni di memorizzazione temporanea di tutto ciò che viene trasferito da e su dischi. A causa della lentezza dei drive, non è possibile, ad esempio, memorizzare ogni carattere di un file mentre viene variato, così come, in fase di lettura, non è possibile leggere i caratteri man mano che appaiono sul video. Nella pratica, il trasferimento di dati da e su disco avviene per blocchi: l'ammontare del numero di blocchi trasferiti in una singola operazione aumenta le prestazioni del sistema perché diminuisce il numero di accessi al disco, tuttavia, poiché diminuisce anche la quantità di memoria a disposizione dei programmi, occorre fare una valutazione che tenga conto di questi due aspetti.
Il valore minimo ammesso è 1, il massimo è 100. I valori tipicamente più diffusi oscillano fra 20 e 30.

CODEPAGE

Ambiente: *Entrambi*

Sintassi: CODEPAGE=codice [,codice]

Default: Video e stampante utilizzano il set di caratteri nativo mentre per la tastiera e il formato delle informazioni relative alla nazione occorre riferirsi al comando COUNTRY.

Questo comando permette di definire due gruppi di caratteri che possono poi essere selezionati dal prompt di comando. Lo scopo di questa definizione è di rendere disponibili per video, tastiera e stampante i caratteri particolari di una certa nazione mantenendo ugualmente disponibile i caratteri standard. Il parametro codice può assumere i seguenti valori:

Codice	Nazione
437	U.S.A. IBM Personal Computer
850	Multilingue
860	Portoghese
863	Franco-Canadese
865	Danese e Norvegese

Nell'appendice C sono riportate le tabelle dei caratteri relativi a ciascun codice di CODEPAGE. È comunque importante ricordare che occorre che monitor e stampante siano predisposti per fornire questa doppia interpretazione: per realizzare ciò, il comando DEVINFO (vedi oltre) deve comparire anch'esso in CONFIG.SYS e dev'essere inoltre disponibile il file che contiene le specifiche dei due set di caratteri per ciascun dispositivo.

Comandi correlati: COUNTRY, DEVINFO in CONFIG.SYS, CHCP.

COUNTRY

Ambiente: *Entrambi*

Sintassi: COUNTRY=codice[,pathname]

Default: COUNTRY=001

Questa specifica permette all'elaboratore di adattare alle convenzioni della nazione selezionata tramite codice, i seguenti dati:

- data;
- ora;
- segno separatore dei decimali di un numero;
- sequenza di ordinamento di lettere maiuscole minuscole, numeri e altri segni per il corretto funzionamento del comando SORT;
- particolari caratteri del linguaggio (accenti, dieresi, ecc.).

Il codice di buona parte delle nazioni è quello corrispondente al prefisso telefonico internazionale. La corrispondenza fra questo codice e il tipo di convenzioni del paese da esso specificato sono memorizzate nel file COUNTRY.SYS che si trova nella root directory del nostro disco e che viene consultato dal sistema operativo quando, trattando il file CONFIG.SYS, incontra il comando COUNTRY, a meno che, nel comando, non sia stato specificato il nome di un altro file nel qual caso sarà quest'ultimo a venir consultato. La tabella seguente illustra la corrispondenza fra codice, nazione e la coppia di valori che per ciascuna nazione possiamo assegnare a CODEPAGE (vedi sopra):

Codice	Nazione/i	CODEPAGE
001	Stati Uniti, Canada (lingua inglese)	437,850
002	Canada (lingua francese)	863,850
003	America Latina	437,850
031	Olanda	437,850
032	Belgio	437,850
033	Francia	437,850
034	Spagna	437,850
039	Italia	437,850
041	Svizzera	437,850
044	Gran Bretagna	437,850
045	Danimarca	865,850
046	Svezia	437,850
047	Norvegia	865,850

049	Germania	437,850
061	Australia	437,850
351	Portogallo	860,850
358	Finlandia	437,850

Comandi correlati: DATE, DIR, KEYB, SORT, TIME

DEVICE

Ambiente: *Entrambi*

Sintassi: DEVICE=pathname[opzioni relative ai diversi device]

Default: I device driver per tastiera, monitor, stampante, dischi e orologio interno vengono comunque caricati dal sistema operativo all'accensione.

Grazie a questo comando la gestione dei dispositivi standard di OS/2 (tastiera, monitor, dischi, stampanti e orologio interno) può essere estesa ad altri dispositivi aggiunti successivamente. Il pathname del comando DEVICE specifica il nome di un file che contiene le istruzioni che mancano al sistema operativo per utilizzare le risorse di un dispositivo diverso da quelli standard: l'insieme delle istruzioni necessarie per ogni dispositivo è detto "device driver". In definitiva, per utilizzare un dispositivo occorre, oltre naturalmente al dispositivo stesso, che il calcolatore possieda una "porta" a cui connettere il dispositivo e per cui il sistema operativo abbia una apposita designazione fra i device (COM1, LPT2, ecc.), inoltre occorre che il sistema operativo conosca le istruzioni per farlo funzionare secondo le aspettative (device driver). Insieme a OS/2 vengono forniti una serie di device driver che, come già i comandi, possono essere utilizzati in ambiente DOS, OS/2 o entrambi: sono ANSI.SYS, COM0n.SYS, EGA.SYS, EXTDSKDD.SYS, MOUSE*sigla*.SYS, POINTDD.SYS e VDISK.SYS (o RAMDISK.SYS). Li esponiamo di seguito sottintendendo per ciascuno di essi l'assegnazione DEVICE= come premessa al device driver:

ANSI.SYS

Ambiente: DOS

Questo device driver permette, durante l'emulazione DOS, di effettuare il controllo del monitor con le sequenze standard ANSI, analogamente a quanto il comando ANSI fà in ambiente OS/2 per il modo protetto.

Comandi correlati: ANSI, PROMPT

COM01.SYS

COM02.SYS

Ambiente: *Entrambi*

Questo device driver permette l'utilizzo della/e porte di comunicazione seriale da parte di programmi come SPOOL (capitolo 5) o altri applicativi generici.
I due diversi file contengono rispettivamente: COM01.SYS le istruzioni necessarie per gestire le due porte seriali della classe di computer AT compatibili; COM02.SYS le istruzioni per gestire le tre porte seriali della classe di computer Personal System/2.

EGA.SYS

Ambiente: DOS

Questo device driver permette l'utilizzo dei registri della scheda Enhanced Graphic Adapter durante l'emulazione DOS. Questa scheda offre una notevole definizione nelle applicazioni grafiche e, in OS/2, è possibile utilizzarla senza nessun device driver aggiuntivo.

Comandi correlati: DEVICE=MOUSEsigla.SYS e DEVICE=POINTDD.SYS in CONFIG.SYS

EXTDSKDD.SYS

Ambiente: *Entrambi*

Questo device driver permette di gestire drive aggiuntivi per dischi esterni e anche di ridefinire i drive che già possediamo identificandoli con un'altra lettera oltre a quella assegnatagli dal sistema operativo. Questo device driver ammette diverse opzioni, di cui solo la prima è richiesta:

/D:numero disco	Rappresenta il numero dell'unità per cui stiamo caricando il device driver: può assumere tutti i valori da 0 a 255. 0 e 1 identificano rispettivamente il primo e il secondo drive a dischi, 128 e 129 il primo e il secondo disco fisso. Le unità esterne, pertanto, avranno come numero minimo 2 se a floppy, 130 se hard disk.
/T:numero tracce	Rappresenta il numero di tracce in cui è suddivisa ciascuna superficie atta alla memorizzazione di dati: può assumere tutti i valori da 1 a 999 e ha valore di default 80.
/S:numero settori	Rappresenta il numero di settori in cui ogni traccia risulta suddivisa: può assumere tutti i valori da 1 a 99 e ha valore di default 9.
/H:numero teste	Rappresenta il numero di testine di lettura/scrittura e quindi di superfici di memorizzazione utili: può assumere tutti i valori da 1 a 99 e ha valore di default 2.
/C	Attiva il controllo sull'apertura del fermo di bloccaggio del floppy nel drive: in questo modo la sola apertura del fermo suddetto forza il sistema operativo a rileggere il disco. Questa opzione ha effetto solo per i calcolatori dotati del drive per dischi apposito.
/N	Specifica che il drive per cui carichiamo il driver è non rimuovibile, cioè un disco fisso.
/F:codice formato	È un numero che indica le caratteristiche del drive al comando FORMAT (capitolo 5) e può assumere i seguenti valori: 0 Floppy disk a 1 testina con capacità 160/180 Kbyte 0 Floppy disk a 2 testine con capacità 320/360 Kbyte 1 Floppy disk a 2 testine con capacità 1,2 Mbyte 2 Floppy disk a 2 testine con capacità 720 Kbyte Il valore assunto per default è 2.

Le lettere A e B sono riservate da OS/2 ai drive interni, siano essi entrambi presenti o meno. Un personal computer dotato di un solo drive ha comunque entrambe le

unità logiche A e B assegnate a quel drive e pertanto il primo drive esterno sarà identificato con la lettera C; se ha un disco fisso, il primo drive esterno sarà identificato con la lettera D. Se noi inseriamo il seguente comando nel file CONFIG.SYS di un elaboratore dotato di un drive a floppy della capacità di 1,2 Mbyte e di un disco rigido, le lettere A e D identificheranno entrambe il drive a floppy:

```
DEVICE=EXTDSKDD.SYS /D:0 /S:15 /F:1
```

MOUSEA00.SYS
MOUSEA01.SYS
MOUSEA02.SYS
MOUSEA03.SYS
MOUSEA04.SYS
MOUSEB00.SYS
MOUSEB01.SYS
MOUSEB02.SYS
MOUSEB05.SYS

Ambiente: *Entrambi*

Questo device driver permette di utilizzare un mouse come dispositivo di puntamento per i programmi dotati di un'interfaccia di questo tipo verso l'utilizzatore. Questo device driver ammette tre opzioni:

MODE=tipo — Questa opzione seleziona l'ambiente operativo in accordo con il tipo:

P =	device driver utilizzabile solo in OS/2;
R =	device driver utilizzabile solo in emulazione DOS;
B =	device driver utilizzabile in entrambi i modi.

B è il valore che viene assunto per default.

QSIZE=numero — Questa opzione specifica il numero massimo di eventi (movimento e pressione dei tasti del mouse) che OS/2 può mettere in attesa prima di effettuarli. Può assumere tutti i valori da 1 a 100 e ha valore di default = 10.

SERIAL=COMn — Questo parametro specifica a quale porta seriale è connesso il mouse. Per la classe di computer AT compatibili i numeri validi sono 1 e 2, per la classe di computer Personal System/2 i numeri validi vanno da 1 a 3. In entrambi i casi, il valore di default è 1. Questo parametro non ha alcun significato per i mouse non connessi a porte seriali.

Poiché molto spesso il mouse è connesso ad una porta seriale, occorre che il device driver opportuno (COM01.SYS o COM02.SYS) sia stato caricato in precedenza, in altri termini, se ad esempio intendiamo installare il mouse su un Personal Computer AT, è indispensabile che il comando DEVICE=COM01.SYS preceda il comando DEVICE=MOUSEA02.SYS (se il mouse che intendiamo installare è, ad esempio, un "Microsoft serial mouse"). I device driver indicati contengono le

istruzioni necessarie al sistema operativo per gestire i mouse dei diversi produttori sulle diverse classi di computer. Per la classe dei computer AT compatibili vale la seguente tabella:

MOUSEA00.SYS	"PC Mouse" prodotto dalla "Mouse Systems": mouse seriale;
MOUSEA01.SYS	"Visi On Mouse" prodotto dalla "VisiCorp": mouse seriale;
MOUSEA02.SYS	"Microsoft Mouse for IBM Personal Computers" prodotto dalla "Microsoft": mouse seriale;
MOUSEA03.SYS	"Microsoft Mouse for IBM Personal Computers" prodotto dalla "Microsoft": mouse parallelo;
MOUSEA04.SYS	"Microsoft mouse for IBM Personal Computers" prodotto dalla "Microsoft": mouse da collegarsi alla tastiera.

Per la classe di computer Personal System/2 vale la seguente tabella:

MOUSEB00.SYS	"PC Mouse" prodotto dalla "Mouse Systems": mouse seriale;
MOUSEB01.SYS	"Visi On Mouse" prodotto dalla "VisiCorp": mouse seriale;
MOUSEB02.SYS	"Microsoft Mouse for IBM Personal Computers" prodotto dalla "Microsoft": mouse seriale;
MOUSEB05.SYS	"IBM Personal System/2 Mouse" prodotto dalla "IBM": mouse da collegarsi alla tastiera (in realtà il mouse si collega ad una presa posta posteriormente al Personal System/2 ma che equivale alla tastiera).

POINTDD.SYS

Ambiente: *Entrambi*

Questo device driver fornisce al sistema operativo il supporto per trasformare le istruzioni provenienti dal device driver di un mouse in un puntatore capace di ottenere effetti grafici sul monitor del computer. le caratteristiche di questo device driver, anche se adatto sia a OS/2 che alla emulazione DOS, sono diverse per i due ambienti e infatti, per ottenere l'utilizzo ottimale del device driver in emulazione DOS, occorre caricare anche EGA.SYS.

Comandi correlati: DEVICE=MOUSE*sigla*.SYS e DEVICE=EGA.SYS in CONFIG.SYS

VDISK.SYS

Ambiente: *Entambi*

Questo device driver permette di utilizzare una certa quantità di memoria RAM come fosse un drive a dischi (disco virtuale): il vantaggio è evidentemente quello di disporre di una enorme velocità di trasferimento deidati, lo svantaggio è nella riduzione della quantità globale di memoria RAM a disposizione. Questo device driver ammette diverse opzioni, elencate di seguito:

dimensione disco	Stabilisce la dimensione massima che avrà il disco in Kbyte: può assumere tutti i valori da 16 a 1024 e ha valore di default 64;
dimemsione settori	Stabilisce il numero di byte per ogni settore: può assumere i valori 128, 256, 512 o 1024 e ha un valore di default di 128;
dimensione root	Stabilisce il numero massimo di "oggetti" che la directory root del disco creato con questo device driver può contenere: può assumere tutti i valori da 2 a 1024 e ha valore di default 64.

Poiché questo opzioni sono posizionali, se occorre specificare, ad esempio, una diversa dimensione della root directory senza specificare gli altri due parametri, questi andranno sostituiti da un virgola. Se, ad esempio, vogliamo dichiarare un disco virtuale con i parametri standard ma con una dimensione della directory root pari a 32, scriveremo:

```
DEVICE=VDISK.SYS ,,32
```

Una nota importante riguarda la sequenza di assegnazione di questo device driver quando è presente anche il device driver EXTDSKDD.SYS: è necessario che quest'ultimo preceda l'assegnazione del disco virtuale per evitare problemi nell'assegnazione delle lettere ai diversi drive.

DEVINFO

Ambiente: *Entrambi*

Sintassi: DEVINFO={KBD|SCR|{PRN|LPTn}}, {device|sigla nazione}, pathname [ROM=codice CODEPAGE,codice tipo caratteri[...]]

Default: Questo comando non ha default in quanto la sua funzione è di mettere i device per i quali viene dichiarato in condizione di gestire i due diversi set di caratteri ed in effetti ha senso solo è preceduto dal comando CODEPAGE (vedi sopra).

Con questo comando si completa l'insieme delle funzioni necessarie per permettere la scelta del set di caratteri da utilizzare per video, tastiera e stampante. Se è presente un comando CODEPAGE, è obbligatorio specificare il comando DEVINFO almeno per video e tastiera; se possediamo inoltre una stampante che permette il doppio set di caratteri, dobbiamo aggiungere un ulteriore comando DEVINFO.
Il comando DEVINFO per la tastiera utilizza il nome KBD, la sigla nazione già vista per il comando KEYB (capitolo 5) e il pathname del file che contiene la codifica dei set (che è obbligatorio fornire in quanto questo comando non ha valori di default), cioè KEYBOARD.DCP. Il comando DEVINFO per il monitor utilizza il nome SCR, il nome del device cui il monitor è fisicamente connesso, cioè EGA per la classe di computer AT compatibili, VGA per la classe di computer Personal System/2 standard e BGA per la classe di computer Personal System/2 dotati della scheda adattatore grafico aggiuntiva e, infine, specifica il nome del file che contiene la codifica dei set, VIOTBL.DCP.
Il comando DEVINFO per la stampante deve individuare a quale, fra le possibili stampanti parallele connesse, intende riferirsi, tramite i nomi LPT1 (equivalente a PRN), LPT2 o LPT3, quindi utilizza il nome del device, attualmente limitato ai due soli modelli di stampanti 4201 e 5202 della IBM, e infine specifica il nome del file che, ancora una volta, contiene la codifica dei diversi set: 4201.DCP se la stampante è una IBM Proprinter o 5202.DCP se la stampante è un IBM Quietwriter.
Per le sole stampanti, è possibile specificare una codifica aggiuntiva del set di caratteri: con la sigla ROM= si dichiara che il set di caratteri è in qualche modo residente all'interno della stampante stessa (Read Only Memory, cartuccia di caricamento formati diversi, ecc.) ed è possibile specificare, per ogni valore della CODEPAGE, un numero identificativo del set.
Facciamo un esempio: supponiamo di avere dichiarato COUNTRY=039 e CODEPAGE=437,850 nel CONFIG.SYS del nostro Personal Computer AT equipaggiato con tastiera italiana, scheda grafica EGA e stampante IBM Proprinter. Come dobbiamo utilizzare il comando DEVINFO? Ecco la risposta:

```
DEVINFO=KBD,IT,KEYBOARD.DCP
```

```
DEVINFO=SCR,EGA,VIOTBL.DCP

DEVINFO=LPT1,4201,4201.DCP
```

Notiamo che i file devono trovarsi, secondo la nostra definizione, della root directory del disco da cui è stato avviato OS/2 (la stessa directory cioè in cui si trova CONFIG.SYS); naturalmente avremmo potuto posizionare i file in un'altra directory e specificarne il pathname completo.

Comandi correlati: CODEPAGE e COUNTRY in CONFIG.SYS, CHCP, KEYB

DISKCACHE

Ambiente: *Entrambi*

Sintassi: DISKCACHE=numero blocchi

Default: DISKCACHE=0

Questo comando offre la possibilità di affidare ad una parte della memoria RAM il compito di velocizzare il trasferimento dei dati da e su disco tramite l'utilizzo diretto di una memoria particolare, detta CACHE (che significa "cassa", per indicare un metodo di trasferimento rapido, qual'è, appunto, il movimento di contanti per cassa), presente in questi elaboratori. Il numero dei blocchi può assumere tutti i valori da 64 a 7200: ogni blocco equivale a 1 Kbyte.

FCBS

Ambiente: DOS

Sintassi: FCBS=numero max FCBS[,numero FCBS non eliminabili automaticamente]

Default: FCBS=16,8

Un FCBS (*File Control Block*) è una sorta di memorandum che il sistema operativo tiene per ogni file che un determinato programma utilizza (si dice che il file è "aperto"). Praticamente, se noi supponiamo di utilizzare un programma applicativo (ad esempio, emissione fatture di vendita) che trae i dati che ci mostra sul monitor da un certo numero di file (anagrafica clienti, anagrafica articoli) per poi aggiornarne altri (fatture clienti, portafoglio, scadenziario attivo), il sistema operativo mantiene tanti FCBS quanti sono i file aperti dall'applicativo. Se il numero di file aperti contemporaneamente supera il numero di FCBS dichiarati con questo comando, il sistema operativo "chiude" quelli aperti più di recente: se raggiunge il limite minino, manda un messaggio di errore sul monitor. Il numero massimo di FCBS può assumere qualsiasi valore da 1 a 255, quello dei file che possono essere chiusi automaticamente può assumere tutti i valori da 0 a 255. Questo comando, residuo del DOS, è stato incluso per compatibilità con programmi già esistenti e va gradualmente eliminato.

IOPL

Ambiente: OS/2

Sintassi: IOPL={YES|NO}

Default: IOPL=NO

Il livello di privilegio assegnato ad un programma determina le risorse del sistema operativo cui il programma può accedere. Ai programmi applicativi che si servono dei device driver del sistema operativo per accedere ai diversi device (disco, stampante, ecc.) viene assegnato normalmente il livello più basso, quello che appunto delega lo scambio di dati con i device al sistema operativo e limita il programma alla propria area dati. Ai programmi dotati dei device driver necessari per comunicare direttamente con un device, viene assegnato un livello di privilegio superiore, tuttavia, affinché il trasferimento di dati da e per un determinato device sia possibile, il comando IOPL deve essere posto uguale a YES. In altri termini, porre a YES il comando IOPL è utile solo se il programma che intendiamo usare comunica direttamente con un device, com'è ad esempio un programma che comanda direttamente un scheda grafica.

LIBPATH

Ambiente: OS/2

Sintassi: LIBPATH=pathname[;pathname][...]

Default: LIBPATH={A|C}:\ (Dipende cioè dal disco da cui è avviato OS/2)

Questo comando stabilisce uno o più path di directory dove il sistema operativo può trovare i file richiesti da un programma in fase di esecuzione. OS/2 permette di realizzare programmi contenenti solo la struttura generale di un comando senza includere in essi tutte le funzioni di servizio necessarie per l'esecuzione del comando. Una volta in esecuzione, questi programmi ricercano sul disco queste funzioni che solitamente sono raggruppate in un file detto, per questo motivo, libreria di funzioni. Questo metodo operativo, in anglosassone "dynamic link", è utilizzato dal sistema operativo stesso e richiede soltanto che sia nota la directory dove si trovano le librerie di funzioni: questo è lo scopo del comando LIBPATH. è da notare che al termine dell'installazione di OS/2 (capitolo 8), il file CONFIG.SYS contiene la riga seguente:

```
LIBPATH=C:\;C:\OS2;C:\OS2\INSTALL
```

MAXWAIT

Ambiente: OS/2

Sintassi: MAXWAIT=numero secondi

Default: MAXWAIT=3

Questo comando stabilisce il numero massimo di secondi durante i quali un'unità di trattamento ("thread") può rimanere inattiva: può assumere tutti i valori da 1 a 255.
In un sistema multi task, i programmi usano le risorse della CPU per pochi millisecondi di seguito in quanto tutti devono usufruire di dette risorse. Un programma, tra l'altro, non è l'entità che sfrutta direttamente queste risorse: è un modo convenzionale con cui il sistema operativo rappresenta l'insieme delle istruzioni, dei dati e dell'utilizzo dei device fatti da un particolare comando.
Il microprocessore del nostro elaboratore dialoga in realtà con i "thread": questa è l'unita di trattamento che effettivamente usufruisce delle risorse della CPU. È evidente che ad un programma in fase di esecuzione corrisponde sempre almeno un thread, niente impedisce tuttavia che un singolo programma possa avviare più thread per utilizzare il microprocessore in modo più efficiente (e il supporto che OS/2 dà a questo metodo di programmazione suggerisce uno sviluppo in questa direzione).
I thread sono classificati secondo tre classi, ciascuna delle quali è suddivisa in 32 livelli di priorità. Le classi identificano rispettivamente i thread per cui l'utilizzo delle risorse della CPU è critico ("Time critical"), quelli che ne fanno un uso regolare ("Regular") e infine quelli che possono rimanere in attesa ("Idle time"): la prima e l'ultima sono gestite direttamente dal sistema operativo, mentre la classe regolare è quella su cui agisce il comando MAXWAIT.
Ai thread di classe regolare viene assegnato dal sistema operativo un livello base di priorità: trascorsi i secondi indicati da MAXWAIT senza che il thread abbia usufruito delle risorse della CPU, il livello di priorità del thread viene aumentato finché il thread non ottiene l'"attenzione" del microprocessore.

Comandi correlati: PRIORITY e TIMESLICE in CONFIG.SYS

MEMMAN

Ambiente: OS/2

Sintassi: MEMMAN={SWAP|NOSWAP},[MOVE|NOMOVE]

Default: MEMMAN=NOSWAP,MOVE se OS/2 è stato avviato da disco A;
MEMMAN=SWAP,MOVE se OS/2 è stato avviato da hard disk.

La coppia di parametri può essere specificata in qualsiasi ordine.
Questo comando stabilisce il modo con cui la memoria RAM può venir trattata da OS/2: il parametro MOVE (NOMOVE) abilita (disabilita) la possibilità di effettuare lo spostamento di segmenti di memoria lasciati liberi da programmi terminati al fine di raccoglierli in aree adiacenti per facilitare l'esecuzione di altri programmi; il parametro SWAP (NOSWAP) abilita (disabilita) la possibilità di scaricare su disco segmenti di memoria appartenenti a programmi in esecuzione ma non utilizzati in quel momento per caricarne altri richiesti da programmi in esecuzione e così via.
L'unico motivo per disabilitare queste funzioni di OS/2 può essere la necessità di gestire procedure per le quali i ritardi causati da queste operazioni, anche se dell'ordine di millisecondi, possono essere critici (come ad esempio il controllo di un processo industriale).

Comandi correlati: SWAPPATH in CONFIG.SYS

PAUSEONERROR

Ambiente: *Entrambi*

Sintassi: PAUSEONERROR={YES|NO}

Default: PAUSEONERROR=YES

Questo comando provoca (evita) l'arresto del trattamento delle istruzioni del file CONFIG.SYS da parte del sistema operativo ogni qual volta un comando risulta errato per qualsiasi ragione (errata sintassi, errato numero e tipo di parametri, errata disposizione rispetto ad altri comandi già eseguiti o da eseguire). Quando questo comando è abilitato, immediatamente dopo il messaggio di errore generato dal comando errato, il sistema si ferma richiedendo all'utilizzatore di premere ENTER, dopodiché riparte, chiaramente senza aver eseguito il comando errato, come peraltro avviene anche quando questo comando è disabilitato.

PRIORITY

Ambiente: OS/2

Sintassi: PRIORITY={DYNAMIC|ABSOLUTE}

Default: PRIORITY=DYNAMIC

Questo comando è in stretta relazione con il comando MAXWAIT visto precedentemente.
Se la priorità dei thread regolari è dinamica, allora vale quanto detto per MAXWAIT a proposito dell'incremento di priorità assegnato dal sistema operativo ad un thread che è in attesa di usufruire delle risorse del microprocessore da un numero di secondi superiore a quello stabilito appunto con MAXWAIT.
Se la priorità è assoluta allora il thread manterrà la priorità che gli è stata assegnata indipendentemente dalla frequenza con cui riuscirà ad usufruire delle risorse del microprocessore.

Comandi correlati: MAXWAIT e TIMESLICE in CONFIG.SYS

PROTECTONLY

Ambiente: *Entrambi*

Sintassi: PROTECTONLY={YES|NO}

Default: PROTECTONLY=NO

Se a PROTECTONLY viene assegnato il valore YES, non sarà più possibile utilizzare l'emulazione DOS: questo significa che potranno essere utilizzati solo programmi eseguibili in modo protetto. A questo proposito è bene prendere qualche precauzione prima di fare una modifica del genere in quanto EDLIN, l'unico editor fornito con OS/2, funziona solo in modo reale e pertanto non sarebbe più possibile utilizzarlo una volta cambiato il valore di PROTECTONLY in YES (la cosa migliore è tenere una copia di CONFIG.SYS, chiamata magari CONFIG.ORG, dove il valore di PROTECTONLY è YES).

Comandi correlati: PROTSHELL, RMSIZE e SHELL in CONFIG.SYS

PROTSHELL

Ambiente: OS/2

Sintassi: PROTSHELL=pathname[argomento|parametro][...]

Default (versione distribuita da IBM):
PROTSHELL=DMPC.EXE SHELL11F.CNF SHELL11F.EXE CMD.EXE

Default (versione distribuita da Microsoft):
PROTSHELL=SHELL.EXE CMD.EXE

Questo comando definisce l'interfaccia utente utilizzata nel modo protetto. pathname identifica un file che contiene questa interfaccia; argomenti e parametri si riferiscono ad eventuali file di configurazione, oltre ad individuare il command processor (cioè il programma che interpreta i comandi digitati in risposta al prompt) e fornirgli le opzioni appropriate.
Al termine dell'installazione di OS/2, versione IBM, il comando PROTSHELL si presenta così:

PROTSHELL=DMPC.EXE SHELL11F.CNF SHELL11F.EXE CMD.EXE /K OS2INIT.CMD

Al termine dell'installazione di OS/2, versione Microsoft, il comando PROTSHELL si presenta così:

PROTSHELL=SHELL.EXE CMD.EXE /K OS2INIT.CMD

Questo comando permette di cambiare il modo con cui OS/2 si presenta al momento dell'accensione: attualmente il program selector e il command processor CMD.EXE sono l'unica interfaccia a disposizione. Questa interfaccia sarà sostituita da Window Presentation Manager (un programma di interfaccia grafico realizzato dalla Microsoft sulla base del proprio applicativo Windows), non appena sarà disponibile la versione 1.1 di OS/2, attesa per l'ultimo quadrimestre del 1988.

Comandi correlati: PROTECTONLY, RMSIZE e SHELL in CONFIG.SYS

REM

Ambiente: *Entrambi*

Sintassi: REM commenti a formato libero

Questo comando permette di inserire commenti all'interno del file CONFIG.SYS: la sua utilità diventa immediatamente chiara quando, dopo diverso tempo, ci si accinge ad intervenire su detto file e, se non si è provveduto a commentare i commandi immessi precedentemente, non si ricordano più parametri e sintassi.

RMSIZE

Ambiente: DOS

Sintassi: RMSIZE=dimensione memoria

Default: RMSIZE={512|640|ammontare totale memoria RAM - 512}

Questo comando specifica quanta memoria RAM riservare per l'emulazione DOS ed ha chiaramente effetto solo se PROTECTONLY=NO. Dimensione memoria può assumere tutti i valori da 0 a 640, con unità di misura uguale a 1 Kbyte. Se la dimensione impostata eccede la quantità di memoria disponibile per il modo reale, essa viene ridotta automaticamente da OS/2 al massimo valore assegnabile rispetto all'ammontare totale.
Il valore di default, cioè quello assunto in assenza del comando RMSIZE, viene calcolato da OS/2 prendendo la dimensione totale della memoria base installata (512 o 640 Kbyte), se tuttavia non resta spazio sufficiente per OS/2, la dimensione viene ridotta secondo il criterio esposto precedentemente.

Comandi correlati: PROTSHELL, PROTECTONLY e SHELL in CONFIG.SYS

RUN

Ambiente: OS/2

Sintassi: RUN=pathname[argomento|parametro][...]

Questo comando permette di avviare programmi durante l'accensione del calcolatore (mentre viene trattato CONFIG.SYS). Pathname identifica il nome del programma; argomenti e parametri sono quelli eventualmente richiesti dal programma.
Un programma avviato con questo comando, poiché parte prima che sia attivata l'interfaccia utente (vedi sopra) e la gestione di errori di disco, deve evitare di utilizzare i device video e tastiera finché il program selector non è partito e inoltre deve possedere un device driver interno per la gestione degli errori di disco.
Un tipico esempio di utilizzo del comando RUN è l'avvio del programma che gestisce la coda di richieste di stampa (SPOOL). Supponendo che il file SPOOL.EXE si trovi nella directory \OS2 del disco C: la sua attivazione avviene tramite il comando seguente:

```
RUN=C:\OS2\SPOOL.EXE
```

Comandi correlati: DETACH

SHELL

Ambiente: DOS

Sintassi: SHELL=pathname[argomento|parametro][...]

Default: SHELL=COMMAND.COM /P

Questo comando definisce l'interfaccia utente utilizzata nel modo reale. pathname identifica un file che contiene questa interfaccia (command processor); argomenti e parametri sono quelli eventualmente accettati dal programma. In emulazione DOS, l'interfaccia utente è contenuta nel file COMMAND.COM; è comunque possibile adottare un'interfaccia diversa sostituendo a COMMAND.COM un altro programma specificandone il nome con questo comando. Se, ad esempio, vogliamo DOSPROC.COM, residente nella directory \PROGRAM del disco C, come command processor, dobbiamo inserire in CONFIG.SYS:

```
SHELL=C:\PROGRAM\DOSPROC.COM
```

Occorre notare che la variabile di ambiente COMSPEC (capitolo 7) va adeguata a questa diversa disposizione ponendola uguale al pathname del nuovo command processor. Nel nostro esempio dovremo inserire la seguente linea nel file AUTOEXEC.BAT (capitolo 7):

```
SET COMSPEC=C:\PROGRAM\DOSPROC
```

Comandi correlati: PROTSHELL e RMSIZE in CONFIG.SYS

SWAPPATH

Ambiente: OS/2

Sintassi: SWAPPATH=path

Default: SWAPPATH={A:\|C:\} (Lo swap viene effettuato sulla directory root del disco di avviamento)

Questo comando definisce la directory che il sistema operativo userà come area di appoggio per i segmenti di memoria che vengono scaricati momentaneamente su disco. Nello specificare questa variabile, occorre tener presente che OS/2 necessita di uno spazio minimo di 512 Kbyte per effettuare lo swap e quindi il disco contenente la directory individuata da SWAPPATH deve avere spazio libero sufficiente.

Comandi correlati: MEMMAN in CONFIG.SYS

THREADS

Ambiente: OS/2

Sintassi: THREADS=numero massimo di thread contemporaneamente attivi

Default: THREADS=64

Queste comando definisce il massimo numero di thread che possono essere contemporaneamente attivi ad un determinato istante, indipendentemente dalla loro classe di appartenenza. Numero può assumere tutti i valori da 32 a 255; il valore di default è suddiviso in 24 threads riservati a OS/2 e i restanti 40 threads a disposizione degli applicativi. Questo numero è chiaramente funzione dei programmi in esecuzione nonché della loro complessità e va aumentato se il numero di thread complessivi attivati dai programmi stessi più quelli richiesti al sistema operativo per la loro esecuzione risulta superiore al valore di default.

TIMESLICE

Ambiente: OS/2

Sintassi: TIMESLICE=numero minimo millisec[,numero massimo millisec]

Default: TIMESLICE=32,248

Questi due numeri definiscono l'intervallo di tempo, in millisecondi, in cui le risorse della CPU sono a disposizione di un thread regolare. Numero minimo non può essere inferiore a 32 e, dal momento in cui un thread inizia ad utilizzare le risorse del microprocessore, queste gli verranno garantite per un tempo uguale almeno al valore di questo parametro. Numero massimo non può essere inferiore a numero minimo e non può superare 65536. Questo secondo valore, posto uguale al primo se non specificato, viene usato dal sistema operativo in un algoritmo detto "round robin" per servire in modo il più equo possibile i thread di uguale priorità. È consigliabile lasciare il numero minimo uguale a 32 perché è un valore chiave per OS/2; inoltre, è bene che il numero massimo non scenda al di sotto di 100 (questo per permettere comunque che l'intervallo di tempo in cui la CPU è a disposizione di un thread sia sufficiente ad eseguire almeno alcune istruzioni) e non diventi troppo grande perché causerebbe solo una dispersione delle risorse.

Comandi correlati: PRIORITY e MAXWAIT in CONFIG.SYS

TRACE

Ambiente: *Entrambi*

Sintassi: TRACE={ON|OFF}[codice evento][,codice evento][...]

Default: Se TRACE non è specificato in CONFIG.SYS, nessun evento verrà memorizzato. Tuttavia, se è specificato TRACEBUF, allora la memorizzazione della successione di eventi può essere comandata dal comando TRACE al prompt di OS/2.

Questo comando permette di memorizzare in un buffer temporaneo il susseguirsi degli eventi mentre OS/2 è in esecuzione; e il suo scopo è quello di fornire supporto nella ricerca di eventuali malfunzionamenti del sistema operativo in particolari condizioni di utilizzo. Il parametro ON (OFF) serve chiaramente ad abilitare (disabilitare) la funzione di TRACE, mentre il codice evento stabilisce quale classi di eventi vanno (cessano di essere) memorizzate:

CODICE EVENTO	TIPO EVENTO (PRINCIPALI)
00	Tutte le classi (equivale a non specificare alcun codice)
01	Riservato
02	Transizioni da modo reale a modo protetto e viceversa
03	"Exceptions" del microprocessore
04	"Interrupts" del microprocessore
05 - 15	Riservati

Esiste un significato anche per i codici fino a 255, si tratta comunque di un livello di approfondimento non previsto per questo testo.

Comandi correlati: TRACEBUF in CONFIG.SYS, TRACE, TRACEFMT

TRACEBUF

Ambiente: *Entrambi*

Sintassi: TRACEBUF=dimensione buffer

Default: TRACEBUF=4 (solo se è specificato un TRACE precedente, altrimenti è 0)

Questo comando è complementare al precedente comando TRACE per lo studio di problemi inerenti il funzionamento di OS/2 in condizioni particolari, e permette di stabilire la dimensione dell'area di memoria RAM (buffer) da riservare per la memorizzazione degli eventi e permettere la loro successiva visualizzazione. Dimensione può assumere tutti i valori da 1 a 63 in unità di 1 Kbyte.

Comandi correlati: TRACE in CONFIG.SYS, TRACE, TRACEFMT

A conclusione dell'esposizione dei comandi di configurazione, vediamo un esempio di configurazione tipica: il file CONFIG.SYS di un Personal Computer AT compatibile dotato di mouse seriale Microsoft:

```
BUFFERS=20
COUNTRY=039
DEVICE=C:\OS2SYS\COM01.SYS
DEVICE=C:\OS2SYS\MOUSEA02.SYS
DEVICE=C:\OS2SYS\POINTDD.SYS
LIBPATH=C:\LIB
PROTSHELL=C:\OS2SYS\SHELL.EXE C:\OS2\CMD.EXE /K
C:\OS2\OS2INIT.CMD
RMSIZE=640
SHELL=C:\OS2\COMMAND.COM /P
SWAPPATH=C:\SWAP
```

APPENDICE A

Comandi e funzioni

I COMANDI DI OS/2

Comandi per il solo ambiente protetto OS/2

 Comandi per il solo ambiente di emulazione DOS

Comando	Scopo
ANSI	Abilita/disabilita l'utilizzo delle sequenze di controllo ANSI standard
APPEND	Stabilisce uno o più path per la ricerca dei file non eseguibili
ASSIGN	Associa la lettera identificativa di un drive a un altro, dirottando così le operazioni di INPUT/OUTPUT
ATTRIB	Attiva/disattiva gli attributi archivio e lettura dei file
BACKUP	Effettua la copia di salvataggio di uno o più file su dischi
BREAK	Stabilisce le modalità di interruzione di un programma tramite i tasti Ctrl+Break

CHCP	Mostra/cambia il set di caratteri in uso
CHDIR	Cambia la directory corrente o ne mostra il nome
CHKDSK	Verifica e corregge gli errori su disco
CLS	Pulisce il video
CMD	Attiva un command processor nello stesso screen group
COMMAND	Attiva il command processor DOS
COMP	Confronta il contenuto di due file mostrando le differenze
COPY	Copia e accoda diversi file
CREATEDD	Crea un dischetto dati per effettuare prove
DATE	Mostra e aggiorna la data
DETACH	Attiva un programma in background
DIR	Elenca i file di una directory
DISKCOMP	Confronta il contenuto dei due dischi
DISCKCOPY	Esegue la copia completa di un disco su un altro
DPATH	Definisce uno o più path per la ricerca di file non eseguibili
ERASE	Cancella uno o più file
EXIT	Termina l'interazione con il command processor in uso
FDISK	Crea, cambia, attiva e cancella le partizioni dei dischi fissi
FIND	Ricerca una serie di caratteri in un file di testo

Comando	Descrizione
FORMAT	Formatta dischi
GRAFTABL	Carica il set di caratteri esteso per l'utilizzo del monitor in modo grafico
HELP	Fornisce spiegazioni sui messaggi di errore inviati dal sistema operativo
JOIN	Associa a un drive il nome di una directory
KEYB	Specifica il tipo di layout della tastiera in uso
LABEL	Inserisce e cambia il nome assegnato a un disco
MKDIR	Crea una nuova directory
MODE	Stabilisce alcuni parametri operativi per i device
MORE	Manda l'output di un comando una schermata per volta
PATCH	Permette di effettuare un patch su file binari
PATH	Definisce uno o più path per la ricerca di file eseguibili
PRINT	Gestisce una coda di stampe
PROMPT	Stabilisce il formato del prompt
RECOVER	Tenta di recuperare il contenuto di dischi con settori difettosi
RENAME	Cambia nome a un file
REPLACE	Sostituisce selettivamente i file con altri
RESTORE	Ripristina le copie di salvataggio effettuate con il comando BACKUP
RMDIR	Cancella una directory
SET	Definisce una variabile di ambiente ponendo il suo nome uguale a un valore

SETCOM40 ◔	Abilita/disbilita l'uso di una potra seriale in DOS
SORT	Ordina i dati forniti in input e li mostra sul video
SPOOL ◕	Gestisce le copie di stampa in multitask
START ◕	Avvia un programma OS/2 in una nuova sessione (screen group)
SUBST ◔	Associa una lettera a una directory così da trattarla come un disco
SYS	Trasferisce i file contenenti il sistema operarivo su un disco
TIME	Mostra e aggiorna l'orario
TRACE ◕	Abilita/disabilita il trace degli eventi del sistema operativo
TREACEMENT ◕	Mostra gli eventi memorizzati in ordine cronologico inverso
TREE	Mostra tutti i path del disco
TYPE	Mostra il contenuto di un file di testo
VER	Mostra la versione del sistema operativo
VERIFY	Attiva/disattiva la verifica della copia effettuata su un disco
VOL	Mostra il nome assegnato a un disco
XCOPY	Copia selettivamente file e directory da un disco all'altro

Comando	Scopo
CALL	Avvia un altro file batch
ECHO	Abilita/disabilita la comparsa dei comandi mentre vengono eseguiti
ENDLOCAL ◔	Ripristina i valori delle variabili di ambiente a quelli precedenti il comando SETLOCAL
EXTPROC ◔	Definisce un diverso programma per l'esecuzione dei file bath
FOR	Permette l'esecuzione ripetuta di comandi
GOTO	Prosegue l'esecuzione di un comando solo al presentarsi di una condizione
PAUSE	Mette il file in attesa della pressione di un tasto
REM	Mostra messaggi contenuti nei file batch
SETLOCAL ◔	Definisce il punto da cui le definizioni delle variabili di ambiente varranno solo per quel file batch
SHIFT	Permette di scorrere in avanti i parametri forniti sulla linea di chiamata dei file batch

I COMANDI DEL FILE CONFIG.SYS

Comando	Scopo
BREAK	Stabilisce le modalità di interruzione di un programma tramite i tasti Ctrl+Break
BUFFERS	Definisce il numero massimo di buffer utilizzabile per le operazioni di inpu/output da disco
CODEPAGE	Seleziona i set di caratteri per permettere la scelta successiva
COUNTRY	Definisce i formati delle informazioni dipendenti dalla nozione
DEVICE	Stabilisce nomi e modalità di device da utilizzare
DEVINFO	Prepara i device per la scelta del set di caratteri
DISKCACHE	Stabilisce un ammontare di memoria per utilizzare la cache del disco
FCBS	Stabilisce il numero di file Control Block
IOPL	Permette ai programmi che posseggono i device driver di accedere direttamente a essi
LIBPATH	Stabilisce un path per le librerie di funzioni a caricamento dinamico
MAXWAIT	Stabilisce il limite massimo di attesa per un thread che necessita di usufruire della CPU
MEMMAN	Seleziona il tipo di gestione della memoria e dello swap
PAUSEONERROR	Abilita/disabilita l'arresto del trattamento di CONFIG.SYS in caso di errore
PRIORITY	Seleziona il trattamento della priorità per i threads

PROTECTONLY	Abilita/disabilita il solo modo protetto
PROTSHELL	Specifica l'interfaccia utente e il command processor di OS/2
REM	Permette di aggiungere commenti nel file
RMSIZE	Stabilisce l'ammontare massimo di memoria RAM per il modo reale
RUN	Avvia un programma automaticamente alla partenza del sistema
SHELL	Definisce il command processor per il modo reale
SWAPPATH	Stabilisce il pathname del file di swap
THREADS	Stabilisce il numero massimo di thread attivi contemporaneamente
TIMESLICE	Stabilisce la durata massima di utilizzo continua di CPU
TRACE	Abilita/disabilita il trace degli eventi dei sistema operarivo
TRACEBUF	Definisce la dimensione del bugger di memorizzazione degli eventi

DIFFERENZE DOS E COMPATIBILITY BOX

Alcuni dei comandi noti agli utenti DOS non sono supportati o hanno minori opzioni nel compatibility box. Ecco l'elenco:

DOS	Emulazione DOS in OS/2
APPEND	L'opzione /x non viene accettata
COMP	È possibile confrontare due file di lunghezza diversa
COUNTRY	L'eventuale pathname viene ignorato in emulazione DOS
CTTY	Comando non più supportato
DISKCOMP	Le opzioni /1 e /8 non servono più: DISKCOPM determina automaticamente il formato
DISKCOPY	L'opzione /1 non serve: DISKCOPY determina automaticamente il numero di superfici
EXE2BIN	Comando non più supportato
FILES	Comando ignorato in CONFIG.SYS
FORMAT	Le opzioni /1, /8 e /B non sono più supportate. L'opzione /S crea un numero di file di sistema maggiore di quello del DOS
GRAPHICS	Comando non più supportato
KEYB	Comando supportato solo in OS/2. L'eventuale pathname viene ignorato in emulazione DOS
LABEL	L'opzione per cancellare il nome non è supportata
LASTDRIVE	Comando ignorato in CONFIG.SYS
MODE	• DISPLAY: opzioni SHIFT e /T non supportate • ASYNC: la gestione del time-out sulla porta seriale è realizzata diversamente • L'indirizzamento di una stampante parallela su linea seriale non è più supportato perché questa funzione è realizzata dal comando SPOOL

NLSFUNC	Comando non più supportato
PRINT	Le opzioni /B, /U, /M, /S e /Q non sono più supportate
SELECT	Comando non più supportato
SHARE	Comando non più supportato
SYS	Il trasferimento dei file di sistema con questo comando è possibile solo se il disco di destinazione è stato formattato

FUNZIONI DELLA TASTIERA E DEL MOUSE

PS = significa valido nella maschera PROGRAM SELECTOR

Nota: Quando sono indicati più tasti separati dal segno +, significa che ciascuno di quei tasti va tenuto premuto mentre si preme il successivo; premuto l'ultimo, vanno rilasciati tutti contemporaneamente.

Effetto	Tastiera	Mouse
Movimento cursore	↑ o ↓ o → o ←	
Movimento del puntatore del mouse		Movimento cursore
Selezionare un elemento della lista	Muovere il cursore sull'elemento, poi Enter	Doppia pressione sul tasto 1 del mouse puntato sull'elemento
Avviare l'elaborazione di quanto scelto o digitato	Enter	Pressione sul tasto 1 del mouse puntato a Enter (PS)
Annullare la videata o il comando corrente	Esc	Pressione sul tasto 1 del mouse puntato su Esc o Cancel (PS)
Consultazione lista elementi "guida" (PS)	↑ o ↓	Pressione sul tasto 1 del mouse puntato su ↑ o ↓ (PS)
Cancellare un carattere per volta a sinistra del cursore	← Backspace	
Inserire caratteri in un comando fino alla pressione di un altro tasto (F1, F3, ...)	Ins	
Cancella un carattere per volta da un comando	Del	

Mostra la funzione "guida" se disponibile (PS)	F1	Pressione sul tasto 1 del mouse puntato a F1=Help (PS)
Mostra il comando precedente 1 carattere per volta		
Mostra tutti i caratteri del comando precedente fino a incontrarne uno uguale a quello digitato dopo F2	F2	
Mostra il comando precedente	F3	
Cancella tutti i caratteri del comando precedente fino a incontrarne uno uguale a quello digitato dopo F4	F4	
Aggiorna la lista dei programmi in esecuzione (PS)	F5	Pressione sul tasto 1 del mouse puntato su F5 (PS)
Mostra la lista delle funzioni "guida" se disponibile		
Memorizza il comando in corso di digitazione come se fosse stato eseguito per poterne effettuare variazioni tramite l'uso dei tasti funzione visti finora		
Mostra le funzioni associate ai tasti (PS)	F9	Pressione sul tasto 1 del mouse su F9=Keys (PS)
Aggiornare il PS (PS)	F10	

Passare in successione da uno screen group al successivo	Alt+Esc
Mostra il program selector	Ctrl+Esc o digitare EXIT al prompt di OS/2 (in modo protetto)
Riportare il cursore al punto da cui è iniziata la consultazione di "guida" (PS)	Alt+F6
Arrestare lo scorrimento di dati sul monitor	Pause oppure Ctrl+NumLock
Terminare un comando	Ctrl+Break
Stampare il contenuto del video	Tasto Print Screen o Shift+PrtSc
Reset del sistema	Ctrl+Alt+PrtSc
Attivare la stampa di un programma in esecuzione	Ctrl+Alt+PrtSc

APPENDICE B

Sequenze di escape ANSI

INTRODUZIONE

Una sequenza di escape ANSI è una sequenza di caratteri (che inizia con un carattere di escape) che permette di controllare gli spostamenti del cursore, cancellare determinate zone del video e determinare il tipo di grafica.

Note

Le note che seguono sono valide per tutte le sequenze di escape:

1. *Pn* rappresenta un parametro numerico il cui valore è un numero decimale
2. *Ps* rappresenta un parametro a selezione. Il suo valore è un numero intero che deve essere selezionato in una lista di valori alternativi
3. *Pl* rappresenta il numero di linea del video ed è un numero decimale
4. *Pc* rappresenta il numero di colonna del video ed è un numero decimale
5. Se non è specificato alcun parametro, o se si è introdotto zero, viene assunto il valore di default.

FUNZIONI DI SPOSTAMENTO DEL CURSORE SUL VIDEO

Le sequenze di escape presentate di seguito modificano la posizione del cursore sul video.

Sigla sequenza	CUP (Cursor Position) oppure HVP (Horizontal and Vertical Position)
Sequenza	ESC [*Pl*; *Pc* H ESC [*Pl*; *Pc* f
Commento	Il cursore viene spostato alla linea e alla colonna specificata dal primo e dal secondo parametro. Il valore di default è 1. Se non è specificato alcun parametro, il cursore è spostato alla prima posizione (home).

Sigla sequenza	CUU (Cursor Up)
Sequenza	ESC [*Pn* A
Commento	Sposta il cursore verso l'alto del numero di linee specificato. Se non è specificato alcun parametro, viene assunto 1. Se il cursore è già sulla prima linea, la sequenza non ha effetto.

Sigla sequenza	CUD (Cursor or Down)
Sequenza	ESC [*Pn* B
Commento	Sposta il corsore verso il basso del numero specificato dal parametro. Se non è specificato alcun parametro, viene assunto 1. Se il cursore si trova già sull'ultima riga del video, la sequenza non ha effetto.

Sigla sequenza	CUF (Cursor Forward)
Sequenza	ESC [*Pn* C
Commento	Sposta il cursore verso destra del numero di colonne specificato dal parametro. Se non è specificato alcun paramatro, viene assunto 1. Se il cursore si trova già sull'ultima colonna a destra del video, la sequenza non ha effetto.

Sigla sequenza	CUB (Cursor Backward)
Sequenza	ESC [*Pn* D
Commento	Sposta il cursore verso sinistra del numero di colonne specificato dal parametro. Se non è specificato alcun parametro, viene assunto 1. Se il cursore è già sulla prima colonna a sinistra, la sequenza non ha effetto.

Sigla sequenza	SCP (Save Cursor Position)
Sequenza	ESC [s
Commento	Viene salvata la posizione corrente del cursore. Il valore salvato può essere successivamente ripristinato dalla sequenza RPS (Restore Cursor Position)

Sigla sequenza	RPC (Restore Cursor Position)
Sequenza	ESC [u
Commento	Ripristina la posizione del cursore salvata dalla sequenza SCP

FUNZIONI DI CANCELLAZIONE

Le sequenze che seguono cancellano aree specifiche del video

Sigla sequenza	ED (Erase Display)
Sequenza	ESC [2J
Commento	Il contenuto del video viene cancellato e il cursore è spostato sulla prima posizione del video (home)

Sigla sequenza	EL (Erase Line)
Sequenza	ESC [K
Commento	Viene cancellata la linea corrente a partire dalla posizione (inclusa) del cursore

FUNZIONI GRAFICHE

Sigla sequenza	SGR (Set Graphics Rendition)
Sequenza	ESC [*Ps*; ...; *Ps* m
Commento	Richiama le funzioni grafiche specificate dai parametri. Esse rimangono attive fino a quando viene emessa un'altra sequenza SGR. I valori dei parametri sono i seguenti:

0	tutti gli attributi rimossi (nero su bianco)
1	alta intensità (neretto)
5	lampeggiamento
7	inverte il colore di fondo
8	nascosto
30	foreground nero
31	foreground rosso
32	foreground verde
33	foreground giallo

34	foreground blu
35	foreground magenta
36	foreground verdazzurro (cyan)
37	foreground bianco
40	background nero
41	background rosso
42	background verde
43	background giallo
44	background blu
45	background magenta
46	background verdazzurro (cyan)
47	background bianco
48	esponenti
49	pedici

Sigla sequenza: SM (Set Mode)
Sequenza: ESC [= *Ps* h
Commento: Imposta la larghezza e il tipo di schermo in base ai parametri specificati. I valori dei parametri sono i seguenti:

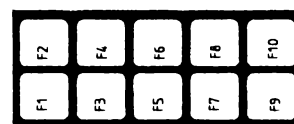
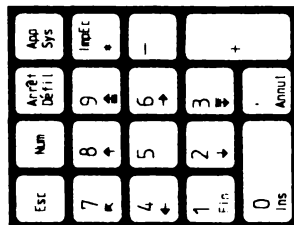
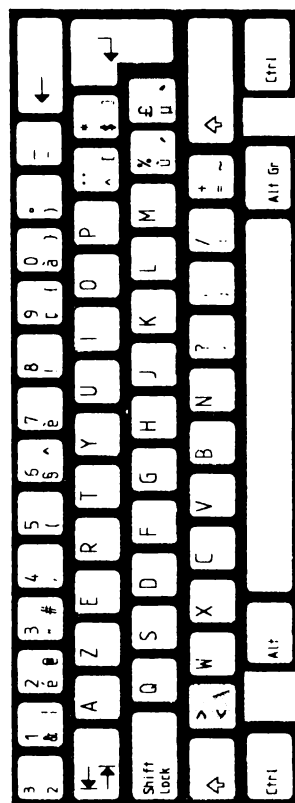
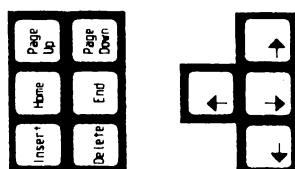
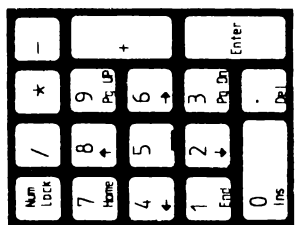
0	40 colonne per 25 linee, biano e nero
1	40 colonne per 25 linee, colori
2	80 colonne per 25 linee, bianco e nero
3	80 colonne per 25 linee, colori
4	320 × 200, colori
5	320 × 200, bianco e nero
6	640 × 200, bianco e nero
7	giunti alla fine di una linea di video, prosecuzione sulla linea di video successiva

Sigla sequenza: RM (Reset Mode)
Sequenza: ESC [= *Ps* 1
Commento: Annulla gli attributi impostati dalla sequenza SM. I valori dei paramatri sono gli stessi della sequenza SM.

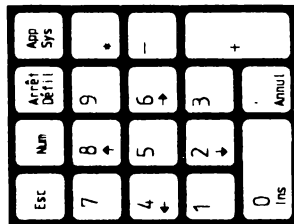
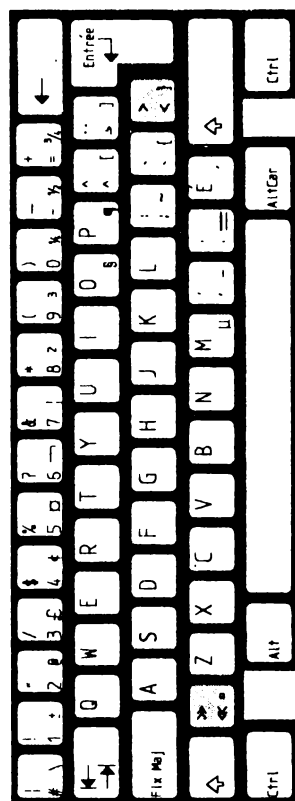
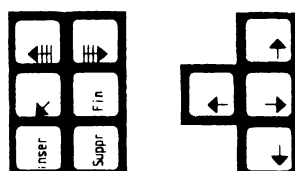
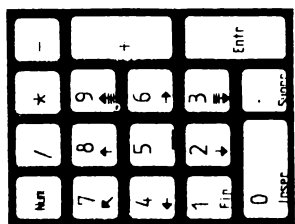
APPENDICE C

Disposizione dei tasti secondo la nazione e set di caratteri

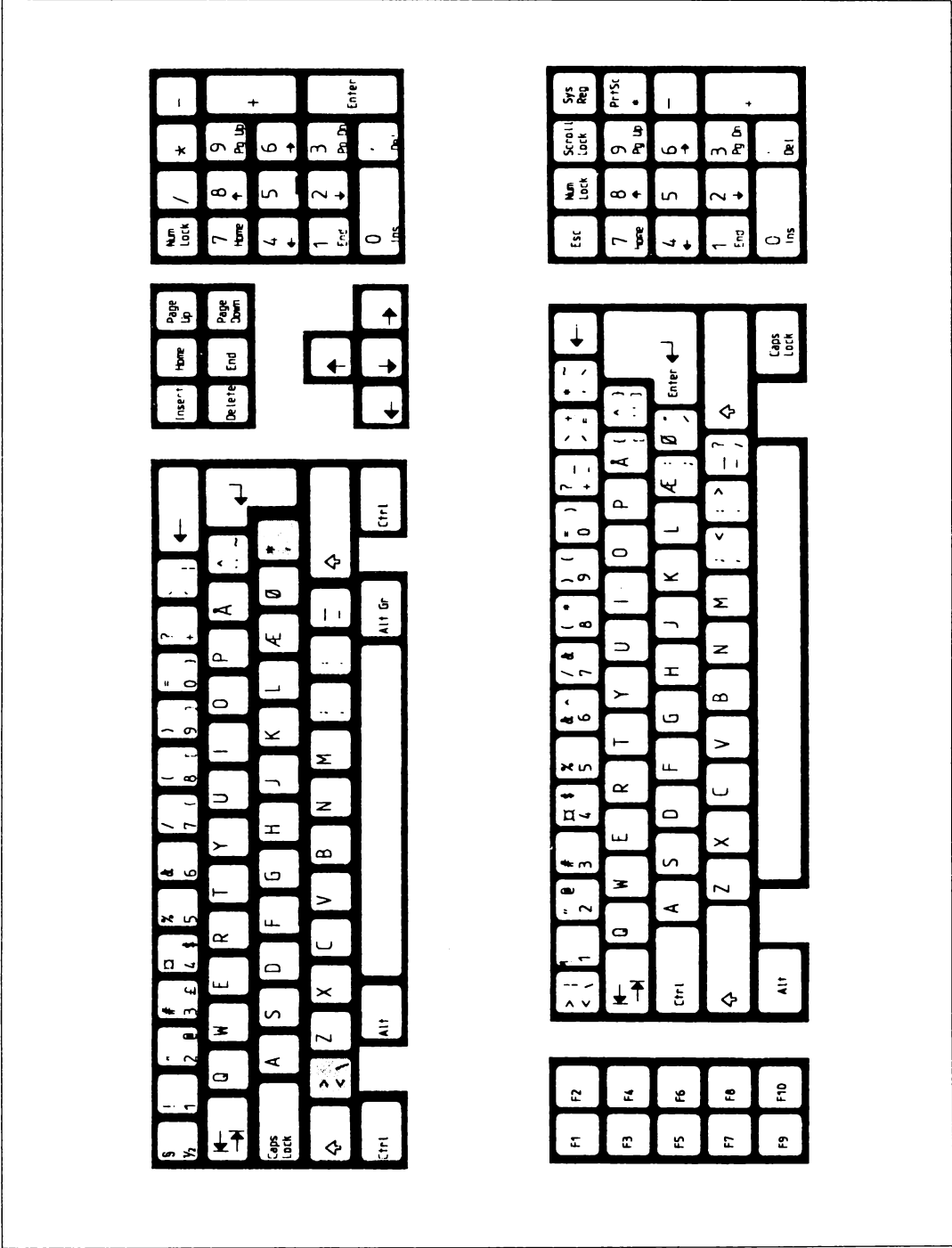
BELGIO

Tastiera avanzata

Tastiera standard

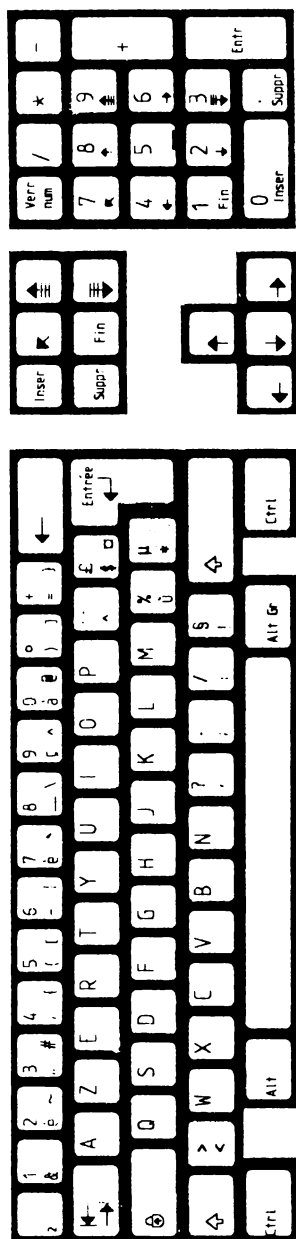
CANADA

Tastiera avanzata

Tastiera standard

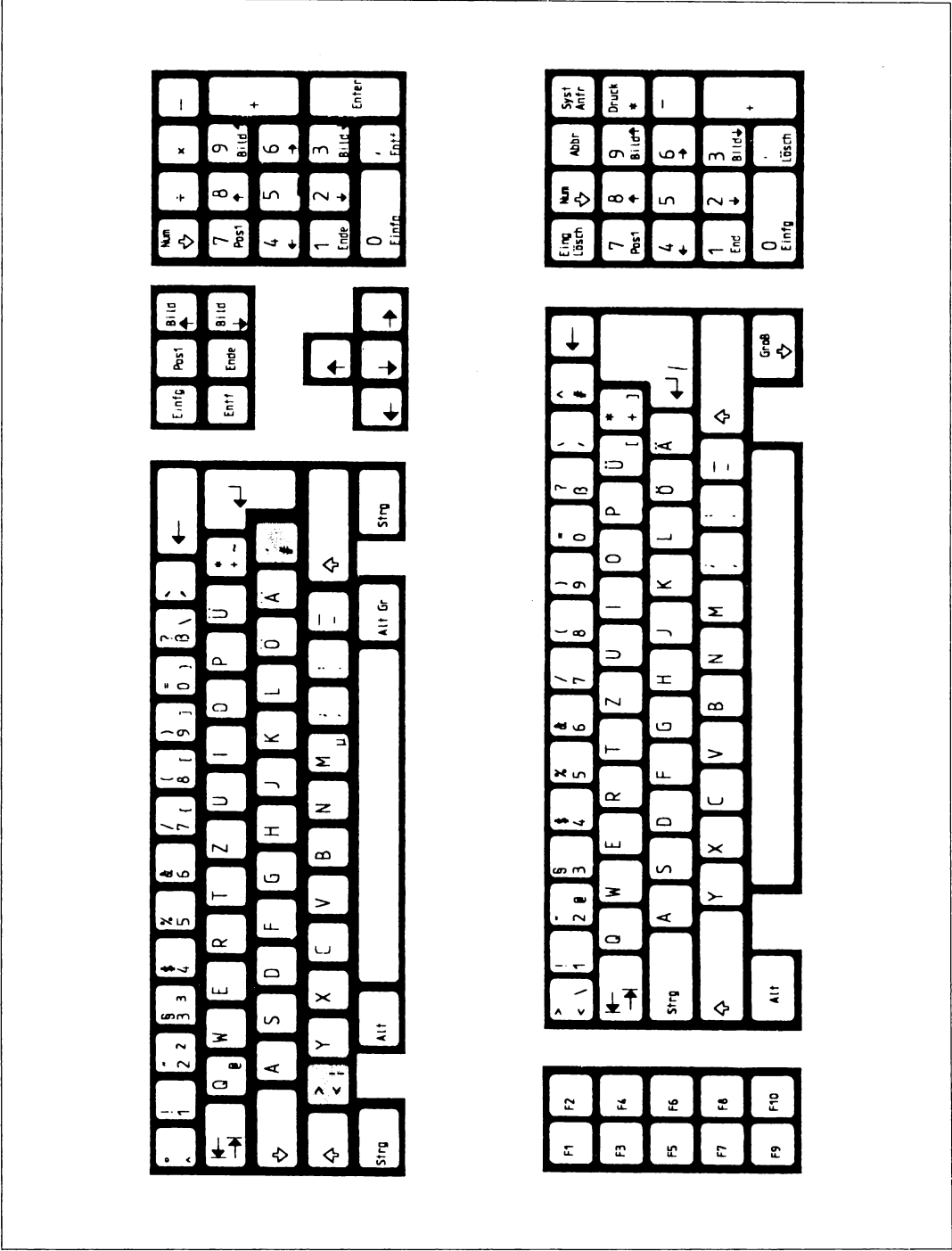
DANIMARCA

Tastiera avanzata

Tastiera standard

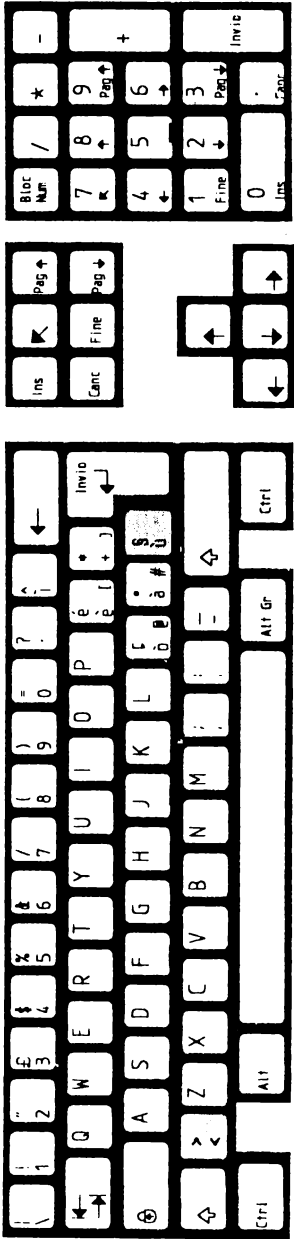
FRANCIA

Tastiera avanzata

Tastiera standard

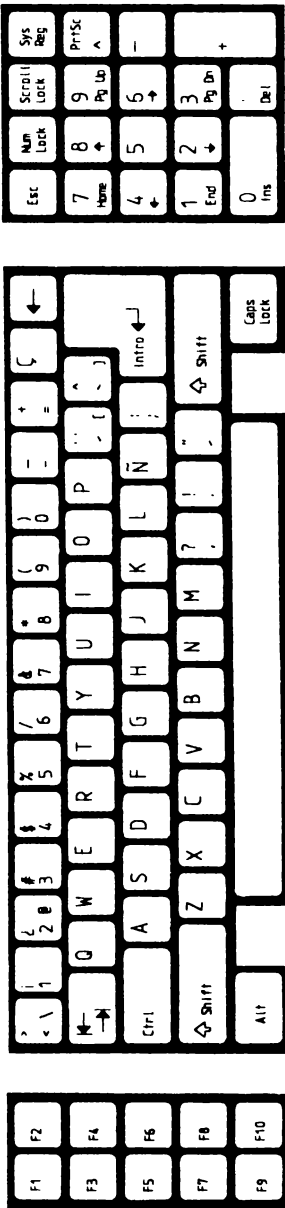
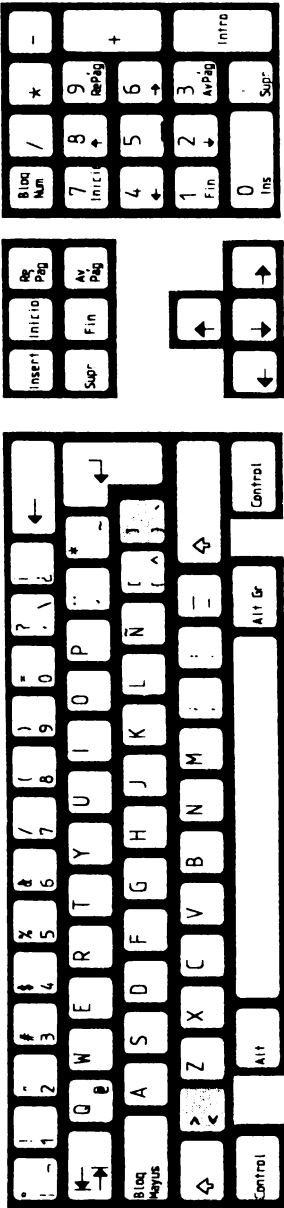
GERMANIA

Tastiera avanzata

Tastiera standard

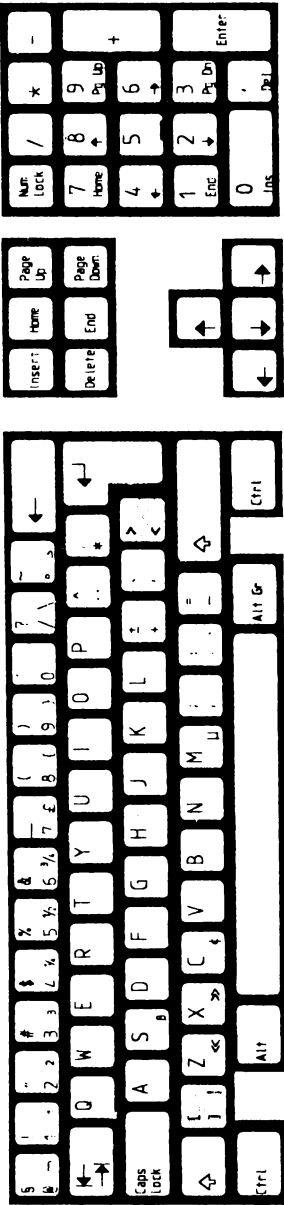
ITALIA

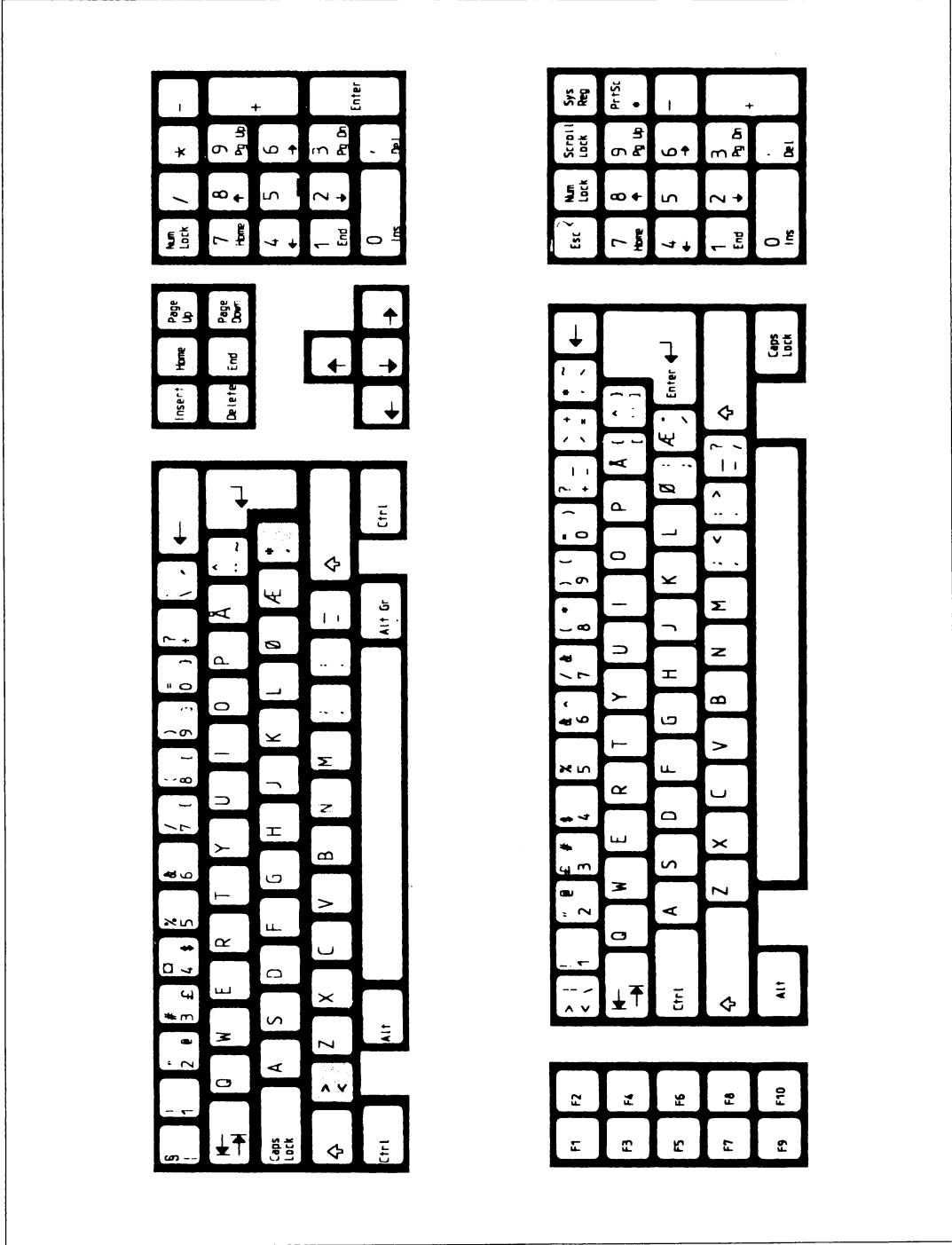
Tastiera avanzata

Tastiera standard

AMERICA LATINA

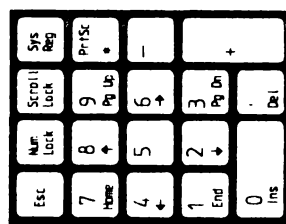
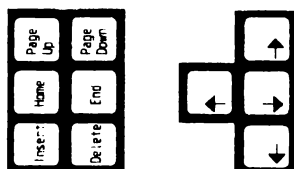
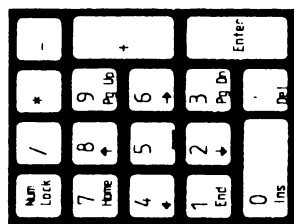
Tastiera avanzata

Tastiera standard

OLANDA

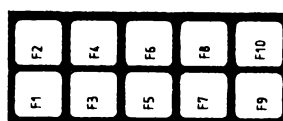
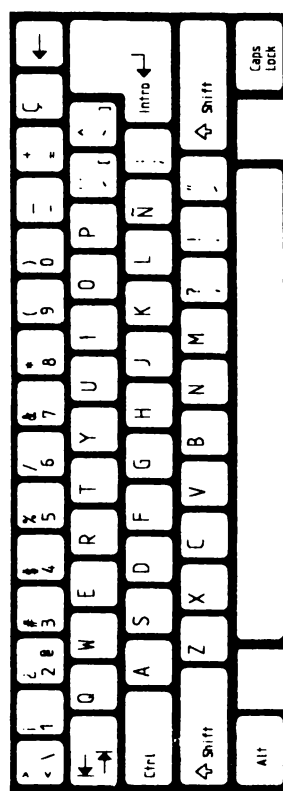
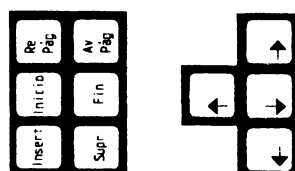
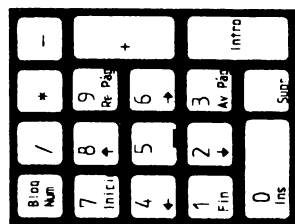
Tastiera avanzata

NORVEGIA

Tastiera avanzata

Tastiera standard

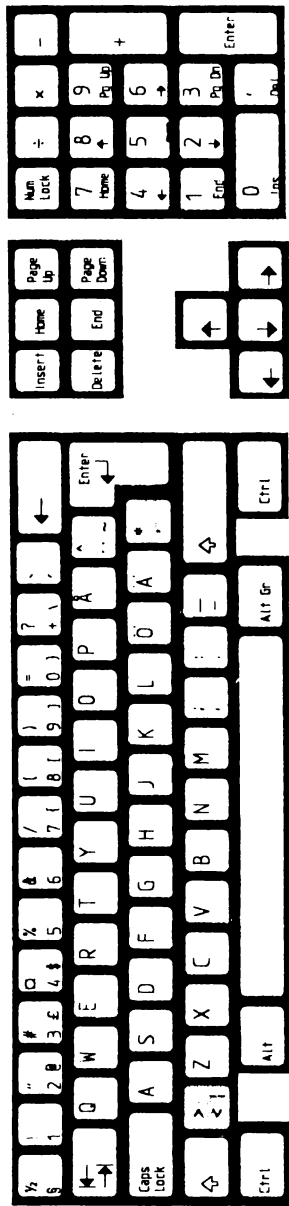
PORTOGALLO

Tastiera avanzata

Tastiera standard

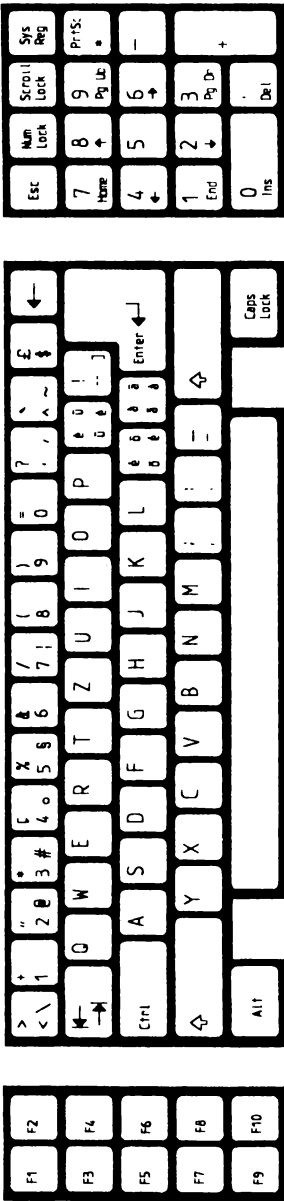
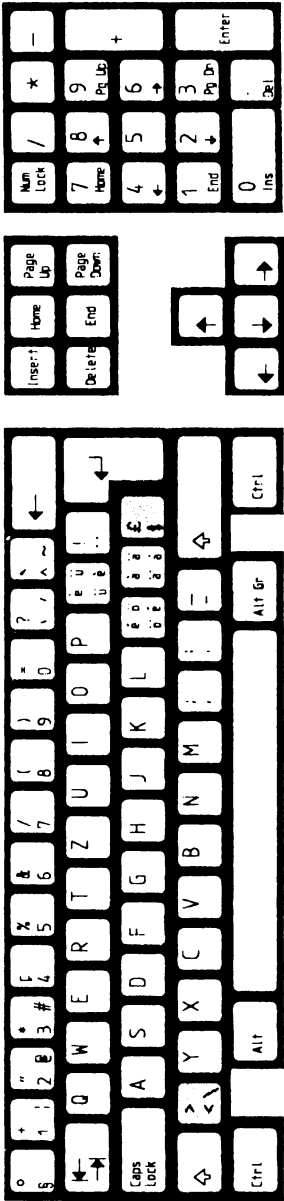
SPAGNA

Tastiera avanzata

Tastiera standard

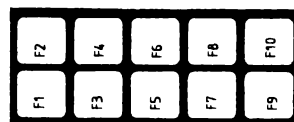
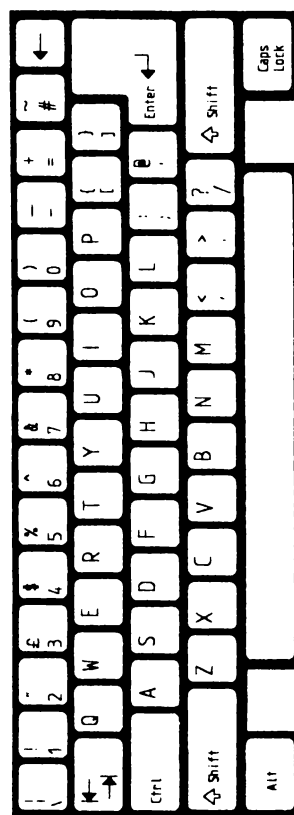
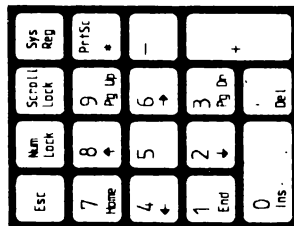
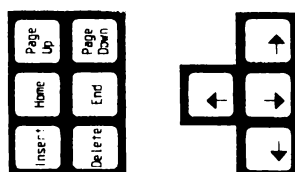
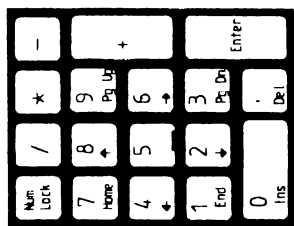
SVEZIA-FINLANDIA

Tastiera avanzata

Tastiera standard

SVIZZERA

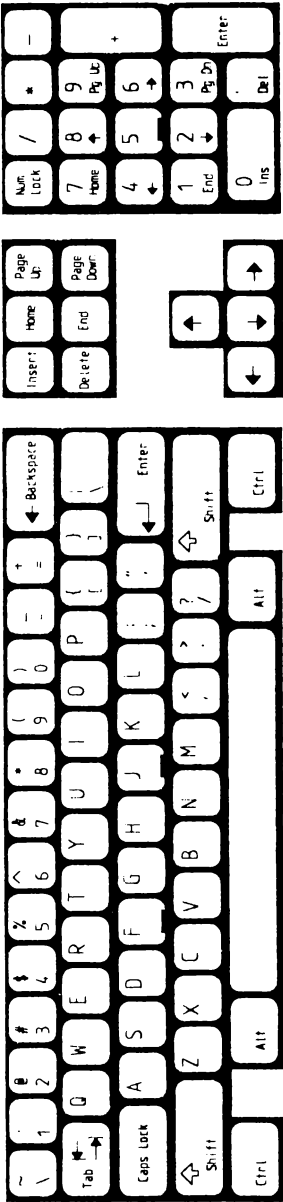
Tastiera avanzata

Tastiera standard

GRAN BRETAGNA

Tastiera avanzata

Tastiera standard

STATI UNITI

Tastiera avanzata

Tastiera standard

CODE PAGE 437 (USA IBM Personal Computer)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	X	p		X	X	X	░	└	╨	a	X
-1	☺	◄	!	1	A	Q	a	q		X	X	I	▒	┴	╤	b	#
-2	☻	↕	"	2	B	R	b	r		X	X	X	▓	┬	╥	Q	/
-3	♥	!!	#	3	C	S	c	s		X	X	X	│	├	╙	f	&
-4	♦	X	$	4	D	T	d	t		X	X	X	┤	─	╘	S	⌠
-5	♣	X	%	5	E	U	e	u		X	X	X	╡	┼	╒	s	⌡
-6	♠	▬	&	6	F	V	f	v		X	X	ª	╢	╞	╓	m	:
-7	•	↨	'	7	G	W	g	w		X	X	º	╖	╟	╫	t	Z
-8	◘	↑	(	8	H	X	h	x		X	X	X	╕	╚	╪	D	°
-9	○	↓	)	9	I	Y	i	y		X	X	⌐	╣	╔	┘	R	•
-A	◙	→	*	:	J	Z	j	z		X	X	¬	║	╩	┌	L	•
-B	♂	←	+	;	K	I	k	X		X	¢	X	╗	╦	█	w	|
-C	♀	∟	,	<	L	\	l	|		X	X	X	╝	╠	▄	B	n
-D	♪	↔	-	=	M	I	m	X		I	¥	!	╜	═	▌	ø	2
-E	♫	▲	.	>	N	^	n	X		X	Pt	''	╛	╬	▐	e	■
-F	☼	▼	/	?	O	_	o	⌂		X	G	''	┐	╧	▀	X	

CODE PAGE 850 (Multilingue)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-		8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p		Ç	É	á	░	└	ð	Ó	-
-1	☺	◄	!	1	A	Q	a	q		ü	æ	í	▒	┴	Ð	ß	±
-2	☻	↕	"	2	B	R	b	r		é	Æ	ó	▓	┬	Ê	Ô	‗
-3	♥	‼	#	3	C	S	c	s		â	ô	ú	│	├	Ë	Ò	¾
-4	♦	¶	$	4	D	T	d	t		ä	ö	ñ	┤	─	È	õ	¶
-5	♣	§	%	5	E	U	e	u		à	ò	Ñ	Á	┼	ı	Õ	§
-6	♠	▬	&	6	F	V	f	v		å	û	ª	Â	ã	Í	µ	÷
-7	•	↨	'	7	G	W	g	w		ç	ù	º	À	Ã	Î	þ	¸
-8	◘	↑	(	8	H	X	h	x		ê	ÿ	¿	©	╚	Ï	Þ	°
-9	○	↓	)	9	I	Y	i	y		ë	Ö	®	╣	╔	┘	Ú	¨
-A	◙	→	*	:	J	Z	j	z		è	Ü	¬	║	╩	┌	Û	·
-B	♂	←	+	;	K	[	k	{		ï	ø	½	╗	╦	█	Ù	¹
-C	♀	∟	,	<	L	\	l	|		î	£	¼	╝	╠	▄	ý	³
-D	♪	↔	-	=	M	]	m	}		ì	Ø	¡	¢	═	¦	Ý	²
-E	♫	▲	.	>	N	^	n	~		Ä	×	«	¥	╬	Ì	¯	■
-F	☼	▼	/	?	O	_	o	⌂		Å	ƒ	»	┐	¤	▀	´	

CODE PAGE 860 (Portoghese)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	X	p	X	X	X	░	└	╨	a	X
-1	☺	◄	!	1	A	Q	a	q	X	X	I	▒	┴	╤	b	#
-2	☻	↕	"	2	B	R	b	r	X	X	X	▓	┬	╥	Q	/
-3	♥	‼	#	3	C	S	c	s	X	X	X	│	├	╙	f	&
-4	♦	X	$	4	D	T	d	t	X	X	X	┤	─	╘	S	⌠
-5	♣	X	%	5	E	U	e	u	X	X	X	╡	┼	╒	s	⌡
-6	♠	▬	&	6	F	V	f	v	X	X	ª	╢	╞	╓	m	:
-7	•	↨	'	7	G	W	g	w	X	X	º	╖	╟	╫	t	Z
-8	◘	↑	(	8	H	X	h	x	X	X	X	╕	╚	╪	D	°
-9	○	↓	)	9	I	Y	i	y	X	X	X	╣	╔	┘	R	•
-A	◙	→	*	:	J	Z	j	z	X	X	¬	║	╩	┌	L	·
-B	♂	←	+	;	K	I	k	X	X	¢	X	╗	╦	█	w	│
-C	♀	∟	,	<	L	\	l	│	X	X	X	╝	╠	▄	B	n
-D	♪	↔	-	=	M	I	m	X	I	X	!	╜	═	▌	ø	2
-E	♫	▲	.	>	N	^	n	X	X	Pt	''	╛	╬	▐	e	■
-F	☼	▼	/	?	O	_	o	⌂	X	X	''	┐	╧	▀	X	

CODE PAGE 863 (Franco-Canadese)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	`	p	X	X	¦	░	└	╨	α	X
-1	☺	◄	!	1	A	Q	a	q	X	X	´	▒	┴	╤	ß	±
-2	☻	↕	"	2	B	R	b	r	X	X	X	▓	┬	╥	Γ	≥
-3	♥	‼	#	3	C	S	c	s	X	X	X	│	├	╙	π	≤
-4	♦	X	$	4	D	T	d	t	X	X	¨	┤	─	╘	Σ	⌠
-5	♣	X	%	5	E	U	e	u	X	X	¸	╡	┼	╒	σ	⌡
-6	♠	▬	&	6	F	V	f	v	X	X	³	╢	╞	╓	µ	÷
-7	•	↨	'	7	G	W	g	w	X	X	¯	╖	╟	╫	τ	≈
-8	◘	↑	(	8	H	X	h	x	X	¤	X	╕	╚	╪	Φ	°
-9	○	↓	)	9	I	Y	i	y	X	X	⌐	╣	╔	┘	Θ	∙
-A	◙	→	*	:	J	Z	j	z	X	X	¬	║	╩	┌	Ω	·
-B	♂	←	+	;	K	[	k	{	X	¢	X	╗	╦	█	δ	√
-C	♀	∟	,	<	L	\	l	|	X	X	X	╝	╠	▄	∞	ⁿ
-D	♪	↔	-	=	M	]	m	}	‗	X	X	╜	═	▌	φ	²
-E	♫	▲	.	>	N	^	n	X	X	X	«	╛	╬	▐	ε	■
-F	☼	▼	/	?	O	_	o	⌂	X	ƒ	»	┐	╧	▀	X	

CODE PAGE 865 (Danese e Norvegese)

Hex Digits 1st → 2nd ↓	0-	1-	2-	3-	4-	5-	6-	7-	8-	9-	A-	B-	C-	D-	E-	F-
-0		►		0	@	P	X	p	X	X	X	░	└	╨	a	X
-1	☺	◄	!	1	A	Q	a	q	X	X	I	▒	┴	╤	b	#
-2	☻	↕	"	2	B	R	b	r	X	X	X	▓	┬	╥	Q	/
-3	♥	‼	#	3	C	S	c	s	X	X	X	│	├	╙	f	&
-4	♦	X	$	4	D	T	d	t	X	X	X	┤	─	╘	S	⌠
-5	♣	X	%	5	E	U	e	u	X	X	X	╡	┼	╒	s	⌡
-6	♠	▬	&	6	F	V	f	v	X	X	ª	╢	╞	╓	m	:
-7	•	↨	'	7	G	W	g	w	X	X	º	╖	╟	╫	t	Z
-8	◘	↑	(	8	H	X	h	x	X	X	X	╕	╚	╪	D	°
-9	○	↓	)	9	I	Y	i	y	X	X	⌐	╣	╔	┘	R	•
-A	◙	→	*	:	J	Z	j	z	X	X	¬	║	╩	┌	L	·
-B	♂	←	+	;	K	I	k	X	X	X	X	╗	╦	█	w	|
-C	♀	∟	,	<	L	\	l	|	X	X	X	╝	╠	▄	B	ⁿ
-D	♪	↔	-	=	M	I	m	X	X	X	I	╜	═	▌	ϕ	²
-E	♫	▲	.	>	N	^	n	X	X	Pt	''	╛	╬	▐	e	■
-F	☼	▼	/	?	O	_	o	⌂	X	G	¤	┐	╧	▀	X	

Glossario

Batch File contenente una serie di comandi che vengono eseguiti dal sistema operativo.

Baud La velocità alla quale i dati vengono trasmessi lungo un canale di comunicazione seriale.

Bit La quantità più piccola che un calcolatore può misurare o riconoscere; corrisponde a una cifra binaria (0 oppure 1). Otto bit equivalgono a un byte.

Byte Unità di misura della memoria RAM del computer e della memoria di massa. Un byte contiene otto bit e può memorizzare un carattere (lettera, numero o segno di punteggiatura).

COM*n* Abbreviazione di Communications. I nomi dei canali di comunicazione seriale.

Compatibili Si dice degli elaboratori che, pur non essendo prodotti dalla IBM (come il PC, il PC XT, l'AT e i nuovi PS/2), offre le stesse caratteristiche e può, di conseguenza, montare lo stesso sistema operativo.

Comunicazione parallela Una tecnica di comunicazione che impiega cavi multipli di collegamento per inviare tutti gli otto bit di un byte contemporaneamente (in parallelo).

Comunicazione seriale Una tecnica di comunicazione che si serve di non più di due fili di collegamento inviando i bit uno dietro l'altro.

CPU Sigla di Central Process Unit (unità centrale di elaborazione). Il termine si usa sia per descrivere la parte di apparecchiatura che contiene il microprocessore, la memoria RAM e gli altri dispositivi fondamentali, sia per individuare il solo microprocessore.

Ctrl-Break Combinazione di tasti che provoca la fine prematura di un comando; si effettua tenendo abbassato il tasto Ctrl e premendo il tasto Break.

Ctrl-C Vedi Ctrl-Break.

Ctrl-NumLock Combinazione di tasti che arrestano l'esecuzione del comando in corso fino a che non viene premuto un altro tasto. Viene di solito utilizzato per fermare lo scorrimento delle linee sullo schermo al fine di permettere la lettura di lunghe videate. Si attiva tenendo abbassato il tasto Ctrl e premendo il tasto NumLock.

Ctrl-P Vedi Ctrl-PrtSc.

Ctrl-PrtSc Combinazione di tasti che controlla la simultaneità di stampa e visualizzazione. Premendo Ctrl-PrtSc una volta il sistema operativo stampa quello che appare sullo schermo; premendolo di nuovo la stampa viene interrotta. Si attiva tenendo abbassato il tasto Ctrl e premendo il tasto PrtSc.

Ctrl-S Vedi Ctrl-NumLock.

Ctrl-Z Combinazione di tasti che crea il carattere speciale utilizzato dal sistema operativo per segnalare la fine di un file. Si ottiene tenendo abbassato il tasto Ctrl e premendo il tasto Z. Si può anche ottenere premendo il tasto funzione F6.

Databit Il numero di bit che costituiscono un carattere nella comunicazione seriale; per rappresentare un carattere vengono generalmente utilizzati otto bit.

Directory corrente È la directory nella quale il sistema operativo effettua le operazioni richieste.

Directory L'indice di un gruppo omogeneo di file su disco. La directory contiene una voce per ogni file, che indica il nome del file, la sua estensione, la grandezza, la data e l'ora in cui è stato modificato. Tutti questi elementi, vengono visualizzati dal comando DIR.

Directory root La directory principale che il sistema operativo crea su ciascun

disco; la directory di livello più elevato in un sistema di gestione dei file a più livelli.

Dispositivo Una parte dell'attrezzatura del computer, come il monitor o la stampante che esegue un compito specifico. Unità periferica.

Drive corrente È il lettore di dischi sui quali il sistema operativo effettua le operazioni richieste.

Drive Unità che permette la lettura e la memorizzazione dei dati.

Enter Tasto che si utilizza per dire al sistema operativo che si ha finito di scrivere una riga.

Esadecimale Il sistema di numerazione a base 16 le cui cifre vanno da 0 a F (le lettere da A a F rappresentano i numeri decimali da 10 a 15). Usato spesso nella programmazione perché è facilmente convertibile da e nel sistema binario, il sistema numerico a base 2 che viene utilizzato dal calcolatore.

Esc Vedi Escape.

Escape Il tasto marcato Esc; cancella una riga che è stata battuta ma non ancora introdotta premendo il tasto Enter.

Estensione Parte finale del nome di un file, quella che segue il carattere punto, lungo non più di tre caratteri.

File di testo Un file che si può leggere (contenente lettere, numeri o segni di interpunzione).

File nascosto Un file che non viene elencato quando si visualizza la directory.

File System Il metodo di organizzazione dei file su disco in una struttura ad albero caratterizzata dalle ramificazioni (directory).

File Un insieme di informazioni registrate su un disco a cui viene assegnato un nome e che contiene dati o un programma.

Filtro Un comando del sistema operativo che legge lo standard input, lo elabora (per esempio ordinandolo alfabeticamente) e scrive il risultato sullo schermo (standard output).

Hardware Le apparecchiature fisiche che compongono un computer, in contrapposizione ai programmi, o software.

I/O Abbreviazione di Input/Output.

Input/Output Termine generico che si riferisce ai dispositivi e ai processi impliciti nella lettura e scrittura di dati da parte del calcolatore.

Interfaccia Confine tra due sistemi o entità, quali ad esempio il drive e il calcolatore, l'utente e il programma.

LPT1,LPT2,LPT3 Abbreviazione di Line Printer, stampante. Nomi utilizzati dal sistema operativo per riferirsi alle tre stampanti parallele che possono essere connesse al calcolatore.

Memoria di massa Supporto per la memorizzazione dei dati. Può essere magnetico o ottico.

Microprocessore Circuito integrato o chip, che contiene i circuiti che effettuano le funzioni di calcolo fondamentali per il calcolatore.

Modo protetto La modalità di funzionamento del microprocessore 80286 che permette il mutitask.

Modo reale La modalità di funzionamento del microprocessore 80286 che emula il funzionamento del suo predecessore 8086 e permette di emulare l'ambiente DOS.

Multitask La possibilità di eseguire più programmi che fruiscono contemporaneamente di tutte le risorse del calcolatore.

Nome dispositivo Il nome con il quale il sistema operativo si riferisce ad un dispositivo (per esempio: PRN, LPT per la stampante). I nomi dei dispositivi vengono considerati dal sistema operativo come nomi di file.

Nome volume Un nome di undici caratteri che si attribuisce a un disco quando lo si formatta.

Nome file Il nome, lungo al massimo otto caratteri, utilizzato dal sistema operativo per catalogare un file su disco.

Parametro Un argomento che si include in una linea di comando per definire più specificamente quello che si vuole che il sistema operativo esegua.

Parità Una tecnica per intercettare gli errori di trasmissione, usata nella comunicazione seriale.

Path La sequenza di nomi di directory che definiscono un determinato percorso.

Pathname Il nome che individua un file percorrendo il path dalla directory root.

Periferiche Vedi dispositivi.

Porta Il collegamento elettrico attraverso il quale il calcolatore invia i dati a, e li riceve da, dispositivi e altri calcolatori.

PRN Abbreviazione di printer (stampante).

Programma applicativo Un programma, per esempio un trattamento testi, che esegue uno specifico compito utile a soddisfare una particolare area o necessità.

Programma Insieme di istruzioni comprensibili al calcolatore.

Prompt Il segno che il sistema operativo visualizza quando è pronto ad accettare un comando.

RAM Abbreviazione di Random-Access Memory (memoria ad accesso dinamico). La memoria utilizzata dal sistema operativo per i programmi e i dati. Il contenuto della RAM varia in fase di utilizzo del calcolatore e si perde tutte le volte che il calcolatore viene spento.

Reindirizzare Dirigere l'output di un comando all'input di un altro comando.

ROM Abbreviazione di Read-Only Memory (memoria a sola lettura). Una parte della memoria del computer è registrata in modo permanente, di solito per contenere un programma. Il contenuto della ROM non si cancella quando si spegne il calcolatore.

Screen group Termine che individua un gruppo di lavoro associato a una determinata maschera video. Può essere costituito da più di un programma (task).

Sessione Usato come sinonimo di screen group, individua il task (o gruppo di task) associati a una determinata maschera video.

Sistema binario Sistema di numerazione a base 2, le cui sole cifre sono 0 e 1. I calcolatori utilizzano il sistema binario poiché le cifre possono essere rappresentate dalla presenza (1) o dall'assenza (0) di tensione.

Sistema Operativo Programma che coordina il lavoro di tutte le parti di un calcolatore.

Software L'insieme di tutti i programmi che permettono e giustificano il funzionamento dell'elaboratore, in contrapposizione al calcolatore vero e proprio, o hardware.

Standard Input Il nome del canale dal quale un programma legge il proprio input a meno che esso non venga reindirizzato; esempio di standard input è la tastiera.

Standard Output Il nome del canale a cui un programma invia il suo output a meno che questo non venga reindirizzato; esempio di standard output è lo schermo.

Stop (bit di) Segnale usato nella comunicazione seriale per indicare la fine di un carattere.

Tasti funzione Tasti recanti la scritta F1, F2 ecc. che fanno svolgere al sistema operativo o a un programma applicativo una serie di funzioni.

Unità centrale Vedi CPU.

Versione Un numero di riferimento che serve per conoscere il livello di aggiornamento del proprio software in relazione alle proposte successive della casa distributrice.

Indice dei comandi

E

F

G

H

I

J

K

L

M

P

Q

R

S

Operating System/2, il nuovo sistema operativo per Personal Computer A.T. compatibili e per il computer della serie Personal System/2, nato dalla statunitense Microsoft in seguito a un progetto comune con IBM, si propone come nuovo standard per le macchine basate sui microprocessori Intel 80286 e 80386.
Questo sistema presenta due caratteristiche importanti e significative, in particolare per chi proviene dall'ambiente MS-DOS: la possibilità dell'esecuzione contemporanea di più procedure, e il superamento del limite della memoria RAM utilizzabile, che passa dal megabyte di MS-DOS ai 16 megabyte indirizzati direttamente da OS/2 sino al gigabyte che OS/2 può indirizzare in modo virtuale. Per mantenere l'enorme volume di software scritto fino ad oggi per MS-DOS, OS/2 offre la possibilità di avere, per una delle maschere video, la compatibilità con il sistema precedente.
Questo volume si propone di essere di aiuto sia a quanti, conoscendo il DOS, decidono di passare a questo nuovo sistema operativo (o semplicemente vogliono conoscerlo meglio proprio per poter decidere), sia a quanti si accingono ad affrontare per la prima volta un sistema operativo.

franco muzzio editore

L. 35.000 ISBN 88-7021-412-5

54 Tiziano Martelli

OS/2 GUIDA ALL'OPERATING SYSTEM 2